BACHELORARBEIT

von Martina Feilke

Sommersemester 2009

SOLAR II - Komplexe Modellierung von beruflichen Allergierisiken und Simulation zur Anwendung von Imputationsmethoden

---

Betreuung:
PD Dr. Christian Heumann
Prof. Dr. Katja Radon, M.Sc.
Jessica Kellberger

Institut für Statistik
Ludwig-Maximilians-Universität München

In Zusammenarbeit mit dem

Institut und Poliklinik für Arbeits-, Sozial- und Umweltmedizin des Klinikums der Universität München

Eidesstattliche Erklärung

Hiermit versichere ich, dass ich die vorliegende Bachelorarbeit selbstständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt habe.

München, den 29. Juni 2009

(Martina Feilke)

Danksagung

Diese Bachelorarbeit entstand am Institut für Statistik der Ludwig-Maximilians-Universität München in Zusammenarbeit mit dem Institut und der Poliklinik für Arbeits-, Sozial- und Umweltmedizin des Klinikums der Universität München.

An dieser Stelle möchte ich mich bei meinen Betreuern PD Dr. Christian Heumann, Prof. Dr. Katja Radon und Jessica Kellberger, die es mir ermöglicht haben an diesem interessanten Thema zu arbeiten, für die freundliche und engagierte Betreuung und die vielen hilfreichen Gespräche und Anregungen bedanken.

Inhaltsverzeichnis

Eidesstattliche Erklärung **3**

Danksagung **4**

1 Einleitung **1**

2 Studiendesign **3**

2.1 Zeitlicher Verlauf der SOLAR-Kohortenstudie 3

2.2 Basiserhebung: ISAAC II 4

2.3 1. Follow-up: SOLAR I 5

2.4 2. Follow-up: SOLAR II 5

2.5 Tätigkeitskodierung und Job-Exposure-Matrix 9

3 Fehlende Daten **12**

3.1 Fehlendmechanismen und grundlegende Begriffe 12

3.2 Umgang mit fehlenden Daten 15

3.2.1 Methoden für Betrachtung der beobachteten Werte 15

3.2.2 Imputationsmethoden - Ersetzen der fehlenden Werte 16

3.2.3 Kombination der Schätzer 20

4 Datenmanagement **22**

4.1 Datengrundlage 22

4.2 Datenbereinigung 23

4.2.1 Korrekturen der Tätigkeitsdaten 23

4.3 Auswahl der Probanden mit vollständigen Tätigkeitsangaben 27

5 Imputation der fehlenden Werte in den potentiellen Confoundervariablen **33**

5.1 Imputation durch Ziehen gemäß der Randverteilung der Daten 38

5.1.1 Binäre Variablen 38

5.1.2 Kategoriale Variablen 38

5.2 Imputation mithilfe des R-Packages AMELIA II 38

5.2.1 Allgemeines . . . 38
5.2.2 Wie funktioniert AMELIA II ? . . . 39
5.2.3 Transformation von Variablen . . . 40
5.2.4 Identifikationsvariablen . . . 41
5.2.5 Auswahl der Variablen bei der Imputation . . . 41
5.2.6 Behandlung von Variablen mit hohen Korrelationen . . . 42
5.3 Übersicht über die Variablenausprägungen in den imputierten Datensätzen 42

6 Berechnung der Expositionsvariablen 44
6.1 Komplexe Matrix als Basis für alle Expositionsberechnungen . . . 44
6.2 Berechnung der Exposition kumuliert über alle Tätigkeiten und Jahre . . 49
6.3 Berechnung der Exposition in der ersten ausgeübten Tätigkeit . . . 49
6.4 Berechnung der Exposition im ersten Tätigkeitsjahr . . . 50
6.5 Betrachtung der gebildeten Expositionsvariablen . . . 52

7 Logistische Regression 60
7.1 Modellannahmen . . . 60
7.2 Parameterschätzung . . . 61
7.3 Parameterinterpretation . . . 61
7.4 Likelihood-Quotienten-Test . . . 63
7.5 Variablenselektion und Modellwahl: AIC-Kriterium . . . 63
7.6 GAM (Generalized Additive Model) . . . 65
7.7 ROC-Analyse . . . 65
7.8 Logistische Regressionsmodelle für die Probanden mit vollständigen Tätigkeitsdaten . . . 67
7.8.1 Mögliche Einflussgrößen (“Confounder”) für die logistischen Modelle 68
7.8.2 Variablenselektion und Modellwahl . . . 69
7.8.3 ROC-Analyse für die “besten” Modelle . . . 77
7.8.4 Schätzer kombinieren . . . 80
7.8.5 Interpretation der Odds-Ratios der kombinierten Parameterschätzer 84
7.8.6 Diskussion der logistischen Regressionsmodelle . . . 87

8 Simulation 90
8.1 Erzeugen eines Fehlendmusters in den Tätigkeitsdaten . . . 92
8.2 Imputation der fehlenden Werte in den Tätigkeitsdaten . . . 94
8.2.1 Vorgehen bei der Imputation . . . 94
8.2.2 Imputation der Zeitangaben . . . 95
8.2.3 Imputation der Wochenstunden . . . 97
8.3 Logistische Regressionsmodelle auf imputierten Tätigkeitsdaten . . . 98

8.4 Vergleich der Parameterschätzer 98

9 Zusammenfassung und Ausblick 106

9.1 Zusammenfassung 106

9.2 Ausblick 107

A Variablenkodierung 109

A.1 Variablen aus ISAAC II 109

A.1.1 In Deutschland geboren 109

A.1.2 Atopie der Eltern 110

A.1.3 Kind gestillt 111

A.1.4 Neurodermitis 112

A.1.5 Allergische Rhinitis 113

A.1.6 Asthma 114

A.1.7 Passivrauch 115

A.1.8 Sozioökonomischer Status 116

A.1.9 Studienzentrum 117

A.1.10 Geschwister 118

A.2 Variablen aus SOLAR I 119

A.2.1 Rauchverhalten 119

A.2.2 Berufssituation 121

A.3 Variablen aus SOLAR II 123

A.3.1 Asthma 123

A.3.2 Allergische Rhinitis 126

A.3.3 Rauchverhalten 128

A.3.4 Berufssituation 129

A.3.5 Schulbildung 131

A.4 Benötigte Variablen für die Tätigkeitsdaten 132

A.4.1 Gearbeitet in SOLAR I 132

A.4.2 Gearbeitet in SOLAR II 133

A.4.3 Gearbeitet in SOLAR I und/oder SOLAR II (unabhängig von der Anzahl der Wochenstunden) 134

A.4.4 Ende der Tätigkeit in SOLAR-I 135

A.4.5 Ende der Tätigkeit in SOLAR II 135

A.4.6 Jemals (mind. acht Wochenstunden) gearbeitet in SOLAR I und SOLAR II 136

A.4.7 Anzahl Tätigkeitsangaben in SOLAR I und SOLAR II 138

A.4.8 Dauer der Tätigkeit 141

A.4.9 Zeilen mit vollständig ausgefüllten Tätigkeitsangaben 141

A.4.10 Probanden mit vollständig ausgefüllten Tätigkeitsangaben 144
A.5 Benötigte Variable für die Simulation 146
A.6 Benötigte Variablen für die Job-Matrix 146
A.6.1 Kurzbeschreibung der in der Basis-Job-Matrix enthaltenen Variablen 146
A.6.2 Kurzbeschreibung der aus der Basis-Job-Matrix gebildeten Variablen 146

B Alle Abbildungen zum Vergleich der Parameterschätzer 147

C R-Code 161
C.1 Imputation der fehlenden Werte in den potentiellen Confoundervariablen . 161
C.1.1 Imputation durch Ziehen gemäß der Randverteilung der Daten . . 161
C.1.2 Imputation mithilfe des R-Packages AMELIA II 166
C.2 Berechnung der Expositionsvariablen 167
C.3 Logistische Regression 175
C.3.1 Schritt 1 - Confoundermodell 175
C.3.2 Schritt 2 - Modelltest 176
C.3.3 Schritt 3 - GAM 176
C.3.4 Schritt 4 - Expositionsvariablen 177
C.3.5 Schritt 5 - Bestes Modell 178
C.3.6 Schritt 6 - Schätzer kombinieren 179
C.4 Simulation 182
C.4.1 Schritt 1 - Werte künstlich löschen 182
C.4.2 Schritt 2 - Imputation der fehlenden Werte in den Tätigkeitsdaten 186

D CD Inhalt 227

Abbildungsverzeichnis

2.1 Zeitlicher Verlauf der SOLAR-Kohortenstudie 3
2.2 Anzahl der Probanden im Verlauf der Kohortenstudie (Stand 31.03.2009) 8
2.3 Job-Exposure-Matrix 11

3.1 Multiple Imputation 19

4.1 Datengrundlage der Analysen 23

5.1 Imputation der fehlenden Werte in den potentiellen Confoundervariablen . 34
5.2 Variablenausprägungen in den imputierten Datensätzen 43

6.1 Beispiel: Tätigkeitsangaben eines Probanden 45
6.2 Beispiel: Komplexe Matrix zur Expositionsberechnung 48
6.3 Beispiel: Exposition kumuliert über alle Tätigkeiten und Jahre 49
6.4 Beispiel: Exposition in der ersten ausgeübten Tätigkeit 50
6.5 Beispiel: Exposition im ersten Tätigkeitsjahr 52
6.6 Boxplots der kumulierten Expositionen auf Basis der vollständigen Tätigkeitsangaben 54
6.7 Boxplots der Expositionen im ersten Tätigkeitsjahr auf Basis der vollständigen Tätigkeitsangaben 56
6.8 Boxplots der Expositionen in der ersten Tätigkeit auf Basis der vollständigen Tätigkeitsangaben 58
6.9 Binäre Expositionsvariblen 59

7.1 Vorgehen bei der Auswahl der logistischen Regressionsmodell für die Probanden mit vollständigen Tätigkeitsangaben 67
7.2 Geschätzte Funktionen für die Expositionsvariablen (Kumulierte Exposition über alle Tätigkeiten und Jahre) 74
7.3 ROC-Kurve für das Modell mit der Zielgröße Allergische Rhinitis in SOLAR II 78
7.4 ROC-Kurve für das Modell mit der Zielgröße Asthma in SOLAR II 79

7.5 Kombination der Parameterschätzer 80
7.6 Konfidenzintervalle der Odds-Ratios - Modell für Allergische Rhinitis in SOLAR II 82
7.7 Konfidenzintervalle der Odds-Ratios - Modell für Asthma in SOLAR II . 84

8.1 Vorgehen bei der Simulation auf jedem der fünf Datensätze mit imputierten Confoundervariablen und vollständigen Tätigkeitsdaten 91
8.2 Imputation des Anfangsjahrs durch Ziehen aus der empirischen Verteilung geschichtet nach dem sozioökonomischen Status 97
8.3 Vergleich der Parameterschätzer - Asthma in SOLAR I 99
8.4 Vergleich der Parameterschätzer - "IRRPEAKS-Exposition kumuliert" bzw. Allergische Rhinitis in SOLAR I 101
8.5 Vergleich der Parameterschätzer - LOWRISK_kumliert 102

B.1 Vergleich der Parameterschätzer - Intercept 148
B.2 Vergleich der Parameterschätzer - Asthma in ISAAC II 149
B.3 Vergleich der Parameterschätzer - Geschlecht 150
B.4 Vergleich der Parameterschätzer - Neurodermitis in SOLAR I 151
B.5 Vergleich der Parameterschätzer - Allergische Rhinitis in SOLAR I 152
B.6 Vergleich der Parameterschätzer - Asthma in SOLAR I 153
B.7 Vergleich der Parameterschätzer - Rauchen in SOLAR I 154
B.8 Vergleich der Parameterschätzer - Sozioökonomischer Status 155
B.9 Vergleich der Parameterschätzer - HMW_kumuliert 156
B.10 Vergleich der Parameterschätzer - LMW_kumuliert 157
B.11 Vergleich der Parameterschätzer - MIXED_kumuliert 158
B.12 Vergleich der Parameterschätzer - IRRPEAKS_kumuliert 159
B.13 Vergleich der Parameterschätzer - LOWRISK_kumuliert 160

D.1 Ordnerstruktur der CD 227

Tabellenverzeichnis

4.1 Korrekturen der Tätigkeitsdaten 26
4.2 Übersicht über die zusätzlich eingeführten “ISCO-Codes” 26
4.3 Übersicht über die Probanden mit vollständigen Tätigkeitsangaben in SOLAR I bzw. SOLAR II 30
4.4 Übersicht über das vorliegende Fehlendmuster 32

5.1 Fehlende Werte: Potentielle Confoundervariablen aus ISAAC II 35
5.2 Fehlende Werte: Potentielle Confoundervariablen aus SOLAR I 36
5.3 Fehlende Werte: Potentielle Confoundervariablen aus SOLAR II 36
5.4 Fehlende Werte: Variablen als Zusatzinformation 37

6.1 Übersicht über die Expositionen über alle Tätigkeiten und Jahre hinweg . 53
6.2 Übersicht über die Expositionen im ersten Tätigkeitsjahr 55
6.3 Übersicht über die Expositionen in der ersten Tätigkeit 57

7.1 Einflussgrößen der Confounder-Modelle für Allergische Rhinitis in SOLAR II 70
7.2 Einflussgrößen der Confounder-Modelle für Asthma in SOLAR II 71
7.3 Einflussgrößen der “besten” Confounder-Modelle 72
7.4 Übersicht über die p-Werte der durchgeführten Likelihood-Quotienten-Tests - Modell für Allergische Rhinitis in SOLAR II 75
7.5 Übersicht über die AIC-Werte der unterschiedlichen Modelle - Modell für Allergische Rhinitis in SOLAR II 76
7.6 Übersicht über die p-Werte der durchgeführten Likelihood-Quotienten-Tests - Modell für Asthma in SOLAR II 76
7.7 Einflussgrößen der “besten” Modelle 77
7.8 Kombinierte Parameterschätzer und Standardabweichungen - Modell für Allergische Rhinitis in SOLAR II 81
7.9 Odds-Ratios und 95%-Konfidenzintervalle - Modell für Allergische Rhinitis in SOLAR II 81

7.10 Kombinierte Parameterschätzer und Standardabweichungen - Modell für Asthma in SOLAR II 83
7.11 Odds-Ratios und 95%-Konfidenzintervalle - Modell für Asthma in SOLAR II 83

8.1 Fehlendmuster (Datensatz mit Tätigkeitsangaben aller Probanden) 92
8.2 Fehlendmuster (Datensatz der Probanden mit vollständigen Tätigkeitsangaben) 93
8.3 Einflussgrößen der Modelle 95
8.4 Einflussgrößen der Modelle bzgl. Zeitangabenimputation 96
8.5 Maximale Abweichung der Odds-Ratios der Schätzer auf den imputierten Tätigkeitsdaten vom Odds-Ratio des“wahren” Schätzers auf den vollständigen Tätigkeitsdaten - Confoundervariablen 104
8.6 Maximale Abweichung der Odds-Ratios der Schätzer auf den imputierten Tätigkeitsdaten vom Odds-Ratio des“wahren” Schätzers auf den vollständigen Tätigkeitsdaten - Expositionsvariablen 105

KAPITEL 1

Einleitung

Allergien und Atemwegserkrankungen werden oftmals durch berufliche Expositionen mitverursacht. Da es zudem immer mehr Menschen gibt, die dazu neigen, allergische Erkrankungen zu entwickeln und im Fall von Berufsasthma schlechte Prognosen bestehen, ist Primärprävention bei Atemwegserkrankungen besonders wichtig. Um entsprechende Präventionsmaßnahmen zu treffen und somit Berufsasthma und Berufsallergien vorzubeugen, müssen jedoch individuelle und berufliche Risikofaktoren bekannt sein. Um Kenntnisse über diese Risikofaktoren zu erlangen, wurde in dieser Bachelorarbeit die SOLAR-Kohortenstudie mit drei Beobachtungszeitpunkten (ISAAC II, SOLAR und SOLAR II) betrachtet, die jeweils am gleichen Kollektiv durchgeführt wurde.

Das Ziel dieser Bachelorarbeit war die Anpassung logistischer Regressionsmodelle für die Zielgrößen “Allergische Rhinitis in SOLAR II” und “Asthma in SOLAR II” an die Daten, wobei die Problematik fehlender Daten berücksichtigt wurde. Außerdem wurde eine Simulation zur Thematik Imputationsmethoden für fehlende Daten und Parameterschätzung durchgeführt.

In Kapitel 2 wird das Studiendesign der SOLAR-Kohortenstudie vorgestellt, wobei genau auf die drei Beobachtungszeitpunkte ISAAC II, SOLAR und SOLAR II eingegangen wird. Außerdem wird die Kodierung von Tätigkeiten und die asthmaspezifische Job-Expose-Matrix beschrieben.

Kapitel 3 handelt vom Umgang mit fehlenden Daten im Allgemeinen.

Die Maßnahmen, die bezüglich des Datenmanagements nötig waren, werden in Kapitel 4 erläutert.

In Kapitel 5 wird beschrieben, wie die Imputation von fehlenden Werten in Confoundervaribalen in dieser Bachelorarbeit durchgeführt wurde. Dabei wird näher auf die beiden Methoden Imputation durch Ziehen gemäß der Randverteilung der Daten und Multiple Imputation mit Hilfe des R-Packages Amelia II eingegangen.

Kapitel 6 erläutert die Berechnung der Expositionsvariablen, die später als Einflussgrößen für logistische Regressionsmodelle dienen sollen.

In Kapitel 7 wird zuerst auf die Theorie zur logistischen Regression und zur Modellwahl eingegangen. Anschließend werden zwei logistische Regressionsmodelle angepasst, deren Findung und Interpretation eines der Hauptziele dieser Arbeit darstellt.

In Kapitel 8 wird eine Simulation zur Anwendung von Imputationsmethoden im Zusammenhang mit der logistischen Regression durchgeführt.

Kapitel 9 fasst die wichtigsten Punkte dieser Bachelorarbeit zusammen und gibt einen kurzen Ausblick.

KAPITEL 2

Studiendesign

2.1 Zeitlicher Verlauf der SOLAR-Kohortenstudie

Der zeitliche Verlauf der SOLAR-Kohortenstudie mit den drei Beobachtungszeitpunkten ISAAC II, SOLAR I und SOLAR II ist Abbildung 2.1 zu entnehmen.

Abbildung 2.1: Zeitlicher Verlauf der SOLAR-Kohortenstudie

In den folgenden Abschnitten werden die zu den drei Beobachtungszeitpunkten durchgeführten Untersuchungen genauer beschrieben. Die Informationen zu diesen drei Beobachtungszeitpunkten wurden größtenteils dem Sachstandsbericht für SOLAR II vom November 2008 [Radon 2008] und dem Abschlussbericht für die SOLAR-Kohortenstudie aus dem Jahr 2005 [Radon 2005] entnommen.

2.2 Basiserhebung: ISAAC II

Das Ziel der ISAAC-Studie ("**I**nternational **S**tudy of **A**sthma and **A**llergies in **C**hildhood") war, die weltweite Prävalenz von Asthma, Allergien und entsprechenden Symptomen im Kindesalter zu beschreiben.

Zu einem ersten Erhebungszeitpunkt 1994/95 wurde mittels eines Elternfragebogens die Prävalenz von Symptomen allergischer und asthmatischer Erkrankungen bei Kindern aus 119 Studienzentren weltweit untersucht.
1995/96 wurde ISAAC II als die zweite Phase der ISAAC-Studie durchgeführt. In der vorliegenden Arbeit werden nur die Probanden aus den beiden Studienorten Dresden und München betrachtet. Für die Auswahl der Probanden wurden zuerst Grundschulen in Dresden und München zufällig ausgewählt. In die Auswahl der für die Studie geeigneten Schulen wurden keine Schulen für körperlich oder geistig behinderte Kinder sowie Schulen mit einem Ausländeranteil von über 80 % aufgenommen, da primär Kinder mit deutscher Nationalität untersucht werden sollten, und so auch Unterschiede genetischer und Lebensstil-Faktoren minimiert werden konnten. Anschließend wurden Probanden aus der Klassenstufe 4 der zufällig ausgewählten Grundschulen zur Teilnahme eingeladen, die zu diesem Zeitpunkt im Alter von 9-11 Jahren waren. Ihre Eltern wurden also vor Beginn der Pubertät ihrer Kinder befragt und die Probanden wurden untersucht.

Die Erhebung umfasste einen ausführlichen Elternfragebogen mit 103 Fragen sowie klinische Untersuchungen in Form von Hautpricktests, Blutuntersuchungen und Lungenfunktionsmessungen. Es wurden 7498 Probanden zur Teilnahme an der Studie eingeladen und von 6399 Probanden liegen von den Eltern beantwortete Fragebögen vor.

Da Dresden in den neuen Bundesländern und München in den alten Bundesländern liegt, konnten zwei genetisch vergleichbare Bevölkerungen, die während der vorangegangenen 40 Jahre verschiedenen Lebensumständen und auch verschiedenen Umweltfaktoren ausgesetzt waren, in Bezug auf Asthma- und Allergieprävalenz miteinander verglichen werden.

Die ISAAC-Studie war ursprünglich als Querschnittsstudie geplant. Es wurde jedoch mit dem Ziel, den Verlauf von Allergien und Asthma und den Einfluss solcher Erkrankungen auf die Berufswahl der Probanden und die Entstehung von Atemwegserkrankungen im Beruf zu untersuchen, 2002/03 eine erste Follow-up-Studie namens SOLAR I durchgeführt.

2.3 1. Follow-up: SOLAR I

Das Ziel der prospektiven, bevölkerungsbezogenen Kohortenstudie SOLAR I (**S**tudie in **O**st- und Westdeutschland zu beruf**l**ichen **A**llergie **R**isiken) war, Jugendliche über den Zeitraum von der Pubertät bis zum Beginn des Berufslebens hinsichtlich des Verlaufs von allergischen Erkrankungen und Atemwegserkrankungen zu beobachten. Dabei sollten vor allem der Einfluss allergischer und asthmatischer Erkrankungen auf die Berufswahl der Jugendlichen und der Einfluss des jeweiligen Berufs auf das Auftreten solcher Erkrankungen untersucht werden. Es sollten außerdem individuelle, für die Entstehung einer Berufsallergie bedeutsame Risikofaktoren gefunden werden.

Hierzu wurden die Teilnehmer der ISAAC II-Studie aus den beiden Studienzentren München und Dresden im Alter von 16-18 Jahren erneut mit einem 121 Fragen umfassenden Fragebogen kontaktiert. Die Befragung fand also vor bzw. zu Beginn des Berufslebens statt. Somit konnten auch Berufswünsche und Ausbildungsziele zeitnah erfragt werden.

5438 der Teilnehmer der ISAAC II-Studie, deren Eltern sich zu einem erneuten Kontakt bereit erklärt hatten, wurden erneut angeschrieben. 4893 Probanden konnten erneut erreicht werden. Davon nahmen 3929 (80,3%) Probanden durch Beantwortung des Fragebogens an der SOLAR I-Studie teil. Davon gaben wiederum 3785 (77,4%) Probanden ihr Einverständnis zur Verknüpfung der neu erhobenen Daten mit den bereits vorhandenen Daten aus der ISAAC II-Studie und ihre Daten konnten somit weiter verwendet werden.

2.4 2. Follow-up: SOLAR II

Die SOLAR II-Studie ist eine Follow-up-Studie von SOLAR I, in der die verbleibenden Fragestellungen aus SOLAR I beantwortet werden sollten.
Da in SOLAR I außerdem bei der Betrachtung der Tätigkeitsangaben die Fallzahlen in den einzelnen Berufsgruppen sehr gering waren, da während des Beobachtungszeitraums dieser Studie erst etwa ein Drittel der Teilnehmer irgendeine Art von Tätigkeit ausgeführt hatte, sollte eine erneute Erhebung durchgeführt werden.

Ziel der SOLAR II-Studie

Durch die SOLAR II-Studie soll eine Optimierung der individuellen Berufsberatung bei atopischen Jugendlichen, die dazu neigen, allergische Erkrankungen zu entwickeln, erreicht werden. Eine solche Optimierung soll dazu führen, dass diese Jugendlichen nicht unnötig von einer großen Anzahl von Berufen ausgeschlossen werden und so zu einer

Senkung der Jugendarbeitlosigkeit beitragen.
Etwa 40% der Jugendlichen sind Atopiker, das heisst sie neigen zur Entwicklung von allergischen Erkrankungen. Diesen Jugendlichen wird derzeit pauschal von Berufen abgeraten, bei denen ein erhöhtes Risiko bekannt ist. Es entwickelt jedoch nur einer von sechs Atopikern auch wirklich eine berufsbezogene Atemwegserkrankung, was bedeutet, dass fünf Jugendlichen unnötig von einem solchen Beruf abgeraten werden muss.
Aus diesem Grund soll ein Punktesystem ("Risikoscore") entwickelt werden, in das neben dem Atopiestatus eines Jugendlichen weitere individuelle Faktoren eingehen sollen. Zu diesen Faktoren zählen die Familienanamnese atopischer Erkrankungen, die Geschwisteranzahl, das Passiv- und Aktivrauchverhalten, ob der Jugendliche als Säugling gestillt wurde, das Vorhandensein von Symptomen in der Kindheit bzw. Jugend und auch berufliche Faktoren wie zum Beipsiel die Betriebsgröße und die Tätigkeitsschwertpunkte im Betrieb.
Mit Hilfe dieses Risikoscores soll das individuelle Risiko eines Jugendlichen, berufsbedingte Atemwegserkrankungen oder Allergien zu entwickeln, vorhergesagt werden. Außerdem kann einem Jugendlichen dann mit erhöhter Vorhersagesicherheit zu einem Beruf geraten oder von einem Beruf abgeraten werden.

Desweiteren soll die Früherkennung von Anzeichen allergischer Atemwegs- und Hauterkrankungen bereits zu Beginn des Berufslebens optimiert werden, damit eventuell erforderliche Arbeitsschutzmaßnahmen rechtzeitig eingeleitet werden können. Somit sollen die jugendlichen Berufstätigen besser vor berufsbedingten Allergien und Asthma geschützt und ihre Arbeitsfähigkeit gesichert werden.
Bei Jugendlichen, die ihre Ausbildung aufgrund von gesundheitlichen Problemen abbrechen, sind in einem Drittel der Fälle Probleme der Haut und der Atemwege ursächlich. Erfolgt ein Berufsabbruch aufgrund von allergischen Erkrankungen oder Asthma, so geschieht das in 88% der Fällen schon während der Ausbildung. Die Folgen eines Berufsabbruchs aufgrund einer Berufskrankheit sind neben gesundheitlichen Problemen auch Umschulungsmaßnahmen. Stehen keine Alternativen in Berufen ohne bekanntes Asthma- oder Allergierisiko zur Verfügung, so kommt es oft sogar zur Arbeitslosigkeit der betroffenen Person.
Deswegen sollen die Nachuntersuchungsintervalle für die Jugendlichen bezüglich allergischer Erkrankungen optimiert werden. Bisher wird empfohlen, eine Nachuntersuchung nach 12 Monaten durchzuführen. Es wäre jedoch möglicherweise vorteilhaft, diese Untersuchung früher durchzuführen.
Auch hierbei soll jedoch nicht jedem Jugendlichen, bei dem zum Beispiel eine Rhinitis neu aufgetreten ist, dazu geraten werden, seinen Beruf aufzugeben, da nicht bei jedem dieser Jugendlichen eine langfristige Atemwegserkrankung entstehen wird. Es soll bei jedem

Jugendlichen mit neu aufgetretenen Problemen der Haut oder der Atemwege individuell entschieden werden können, ob eine Umschulungsmaßnahme dringend erforderlich ist oder ob eine Expositionsminderung bzw. eine intensive Nachbeobachtung ausreichend ist. So soll die Anzahl von Umschulungsmaßnahmen reduziert und eine Senkung des Risikos für Jugendarbeitslosigkeit erreicht werden. Die Prognosesicherheit soll erhöht werden, indem wiederum das individuelle private und berufliche Risikoprofil betrachtet wird.

Auswahl der Probanden

Für die SOLAR II-Studie wurden diejenigen Probanden, die bereits 2002/03 im Alter von 16-18 Jahren an der SOLAR I-Studie teilgenommen hatten, 2007-2009 im Alter von 21-23 Jahren erneut angeschrieben. Die Befragung fand somit zu bzw. nach Beginn des Arbeitslebens statt.
Somit befanden sich zum Zeitpunkt der Befragung nicht nur Haupt- und Realschüler sondern auch Abiturienten bereits in einer Ausbildung oder gingen verschiedenen anderen (Neben-)Tätigkeiten nach. Somit konnte ein größerer Teil des Berufs- und Tätigkeitsspektrums und der Bevölkerung abgedeckt werden als in der SOLAR I-Studie.
Durch die longitudinale Verknüpfung der zu den drei Beobachtungszeitpunkten ISAAC II, SOLAR I und SOLAR II erhobenen Daten erhielt man außerdem die Chance, den Verlauf von Atemwegs- und allergischen Erkrankungen vom Kindesalter über die Pubertät bis zum Eintritt ins Berufsleben zu verfolgen.

Seit dem 18. Juli bzw. 1. August 2007 wurden monatlich 100 bis 150 Probanden in den Studienzentren München und Dresden zur Teilnahme an der Studie eingeladen. Es wurde jeweils ein Fragebogen mit insgesamt 136 Fragen verschickt, zudem wurden die Probanden zu einer klinischen Untersuchung eingeladen.
3054 der Teilnehmer der SOLAR I-Studie konnten erneut für SOLAR II angeschrieben werden, diese hatten sich in SOLAR I zu einer erneuten Kontaktaufnahme bereit erklärt. Zum Zeitpunkt der Abfassung dieser Bachelorarbeit konnten von 1966 Teilnehmern die Daten aus den Studien ISAAC II, SOLAR I und SOLAR II miteinander verknüpft werden (Abb. 2.2).

Anzahl der Probanden

Die Anzahl der Probanden, die in SOLAR II den Fragebogen beantwortet und ihr Einverständnis zur Verknüpfung der Daten mit den Daten aus ISAAC II und SOLAR I gegeben haben, kann nicht als endgültig angesehen werden.

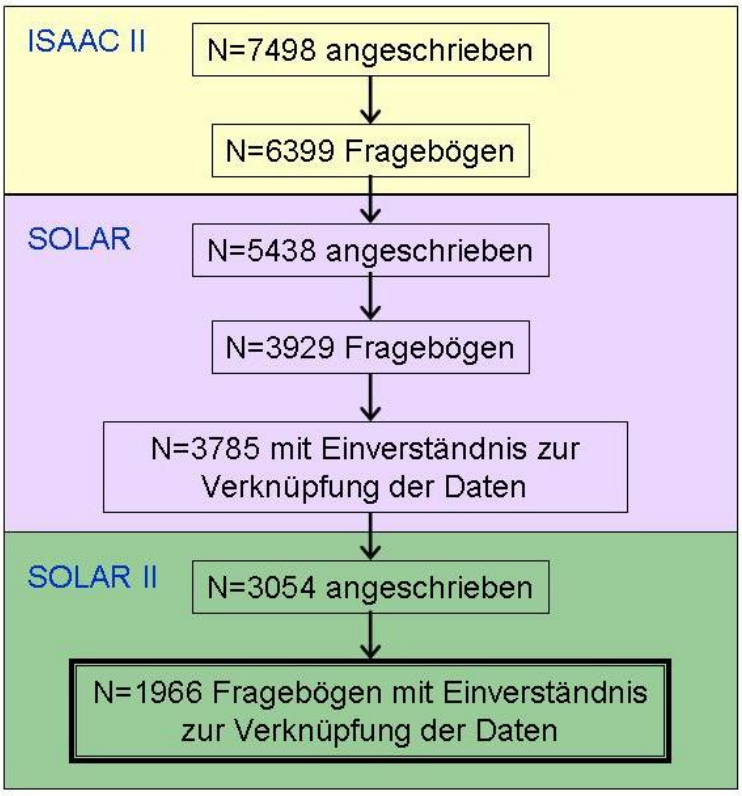


Abbildung 2.2: Anzahl der Probanden im Verlauf der Kohortenstudie (Stand 31.03.2009)

Für diese Bachelorarbeit wurden nur die Daten derjenigen Probanden verwendet, die an allen drei Studien (ISAAC II, SOLAR I und SOLAR II) teilgenommen haben und deren Daten aus SOLAR II zum Zeitpunkt des Beginns dieser Bachelorarbeit bereits mit den Daten aus ISAAC II und SOLAR I verknüpft werden konnten. Außerdem mussten die beruflichen Tätigkeiten der Probanden zum Zeitpunkt des Beginns dieser Arbeit schon kodiert worden sein.

Beteiligte Einrichtungen

Die Durchführung der Studien erfolgte durch das Institut und die Poliklinik für Arbeits-, Sozial- und Umweltmedizin des Klinikums der Universität München in Zusammenarbeit mit dem Dr. von Haunerschen Kinderspital des Klinikums der Universität München, dem Universitätsklinikum Carl Gustav Carus Dresden, der Universität Ulm und der Justus-Liebig-Universität Giessen.

2.5 Tätigkeitskodierung und Job-Exposure-Matrix

Tätigkeitskodierung

In den Studien SOLAR I und SOLAR II wurden von den Probanden Angaben zu ihren beruflichen Tätigkeiten, Ferienjobs, Aushilfstätigkeiten usw. gemacht. Die von den Probanden im Fragebogen angegebenen Tätigkeiten wurden nach der Internationalen Standardklassifikation der Berufe (ISCO-88), die von der Internationalen Arbeitsorganisation (ILO) in Genf entwickelt wurde, kodiert [Geis 2007]. Diese Tätigkeitskodierung dient ursprünglich vor allem dazu, die internationale Vergleichbarkeit von Arbeitsmarktstatistiken zu ermöglichen. Durch das Einteilen der Tätigkeiten in bestimmte Tätigkeitsgruppen durch die ISCO-Klassifizierung kann jeder Tätigkeit ein vierstelliger Zahlencode zugeordnet werden.

Es gibt 10 Hauptgruppen (major groups), in welche die Tätigkeiten der Probanden anhand der ISCO-Kodierung eingeordnet werden konnten:

0. Soldaten
1. Angehörige Gesetzgebender Körperschaften, leitende Verwaltungsbedienstete und Führungskräfte in der Privatwirtschaft
2. Wissenschaftler (=Hochschulabsolventen)
3. Techniker und gleichrangige nichttechnische Berufe (=Fachhochschulabsolventen)
4. Bürokräfte, kaufmännische Angestellte
5. Dienstleistungsberufe, Verkäufer in Geschäften und auf Märkten
6. Fachkräfte in der Landwirtschaft und Fischerei
7. Handwerks- und verwandte Berufe
8. Anlagen- und Maschinenbediener sowie Montierer
9. Hilfsarbeitskräfte

Diese 10 Hauptgruppen werden durch 3 Gliederunsgebenen weiter spezifiziert:

- Hauptuntergruppen (sub-major groups)
- Untergruppen (minor groups)
- Gattungen (unit groups)

Somit konnte jeder Tätigkeit ein vierstelliger Code zugeordnet werden und dieser konnte dann später in die Job-Exposure-Matrix überführt werden.

Ein Beispiel für die Bestimmung des ISCO-Codes (für einen Tierarzt):

2 *Wissenschaftler*
22 *Biowissenschaftler und Mediziner*
222 *Mediziner* (*ohne Krankenpflege*)
2223 *Tierärzte*

Um die Validität der Daten zu erhöhen, wurde die Kodierung von zwei Personen unabhängig voneinander durchgeführt. Bei Differenzen zwischen den beiden Kodierungen wurde die endgültige Kodierung von einem Experten festgelegt.

Job-Exposure-Matrix

Um die für die Asthma- und Allergieentstehung relevanten beruflichen Expositionen bei den von den Probanden angegebenen Tätigkeiten abschätzen und einordnen zu können, wurde die Job-Exposure-Matrix (JEM), die von Dr. Susan Kennedy (University of British Columbia) [Kennedy et al. 2000] entwickelt wurde, verwendet. In der JEM werden die spezifischen Expositionen betrachtet, die zu Berufsasthma oder einer Berufsallergie führen können. Jedem vierstelligen ISCO-Code kann mit Hilfe der JEM eine bestimmte Exposition zugeordnet werden.
Die JEM enthält die durch die ISCO-88-Klassifikation festgelegten ISCO-Codes (zeilenweise) und 22 verschiedene Expositionsgruppen, die zu fünf Expositionskategorien zusammengefasst werden können (spaltenweise). Vier Kategorien der JEM bedeuten ein hohes Asthmarisiko: HMW (Hochmolekulare Stoffe), LMW (Niedermolekulare Stoffe), Mixed (Gemischte Stoffe) und Irrpeaks (Irritative Spitzenexposition). Die fünfte Kategorie bedeutet ein niedriges Asthmarisiko.
Besteht bei einer Tätigkeit eine Exposition in einer dieser fünf Kategorien, so steht in der JEM in der jeweiligen Kategorie eine 1. Anderenfalls, wenn keine Exposition vorliegt, steht eine 0. So wird also nur unterschieden, ob überhaupt eine Exposition vorliegt oder nicht. Es wird aber nicht unterschieden, wie stark, also wie intensiv der jeweilige Proband einem bestimmten Stoff ausgesetzt ist.
Bestand weder ein hohes noch ein niedriges Asthmarisiko, so wurde der Proband als nicht exponiert kodiert und in allen Kategorien eine 0 eingetragen. In einem Expertenschritt wurden von Frau Prof. Dr. Radon die ISCO-Kodierungen und die zugehörigen Expositionsangaben überprüft und falls notwendig korrigiert.

Die Einteilung der beruflichen Exposition anhand der Job-Exposure-Matrix (JEM) wird nachfolgend graphisch dargestellt (Abb. 2.3). Die Erstellung dieser Abbildung erfolgte in Anlehnung an die graphische Veranschaulichung der JEM im Abschlussbericht für die SOLAR-Kohortenstudie aus dem Jahr 2005 [Radon 2005].

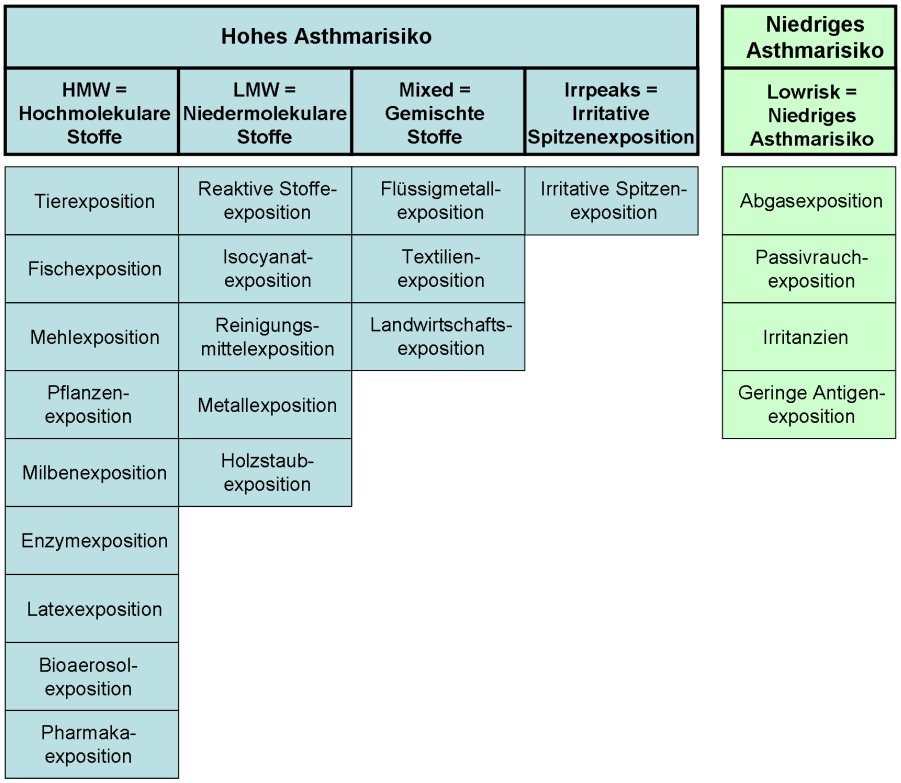


Abbildung 2.3: Job-Exposure-Matrix

KAPITEL 3

Fehlende Daten

Bei der Analyse epidemiologischer Datensätze stößt man fast immer auf das Problem fehlender Daten, das sich darin äußert, dass einzelne Beobachtungen oder Variablen innerhalb eines Datensatzes fehlen.
So kann es zum Beispiel bei klinischen Studien vorkommen, dass Patienten aus der Studie ausfallen. Gründe hierfür können sein, dass sie aufgrund von Nebenwirkungen eines Medikaments nicht mehr an der Studie teilnehmen möchten oder während der Studie an einen anderen Ort ziehen und somit nicht mehr auffindbar sind.
Bei Umfragen und dem Ausfüllen von Fragebögen für klinische Studien kann es zu unvollständig ausgefüllten Fragebögen kommen. Dabei können die fehlenden Antworten zufällig fehlen, weil zum Beispiel eine Frage übersehen wurde. Sie können aber auch nichtzufällig fehlen, weil Probanden zum Beispiel Fragen nach dem Einkommen oder dem Alkoholkonsum nicht beantworten möchten.
Auch in der SOLAR-Kohortenstudie lag das Problem fehlender Daten vor, das sich in den für diese Arbeit vorliegenden Daten durch einzelne fehlende Beobachtungen bemerkbar machte. Diese kamen dadurch zustande, dass Probanden in den Fragebögen zum Teil unvollständige Angaben gemacht haben.

Im Folgenden werden Mechanismen vorgestellt, die zu fehlenden Daten führen können (Fehlendmechanismen). Anschließend werden einige Methoden für den Umgang mit fehlenden Daten vorgestellt.

3.1 Fehlendmechanismen und grundlegende Begriffe

Zur Veranschaulichung der Fehlendmechanismen betrachtet man im einfachsten Fall nur zwei Variablen: Eine Variable Y, bei der n Einheiten vollständig beobachtet wurden, und eine zweite Variable X, bei der f Werte fehlen und für die deswegen nur n-f Werte beobachtet wurden. Folgende Situationen können hier eintreten:

1. Die Wahrscheinlichkeit für das Fehlen der Werte hängt weder von X, noch von Y ab.
2. Die Wahrscheinlichkeit für das Fehlen der Werte hängt von Y, aber nicht von X ab.
3. Die Wahrscheinlichkeit für das Fehlen der Werte hängt von X, aber nicht von Y ab.
4. Die Wahrscheinlichkeit für das Fehlen der Werte hängt von X und von Y ab.

Im ersten Fall spricht man von "missing completely at random" (MCAR). Die beobachteten Daten der Variable X bilden dann eine Zufallsstichprobe aus den gesamten Daten der Variable X, die aus den beobachteten und den fehlenden Daten bestehen. Im zweiten Fall spricht man von "missing at random" (MAR). Die beobachteten Daten der Variable X bilden dann nicht notwendigerweise eine Zufallsstichprobe aus den gesamten Daten der Variable X. Die Fälle 3 und 4 werden beide mit "not missing at random" (NMAR) bezeichnet.

Betrachtet man nun nicht mehr nur zwei sondern mehrere Variablen, z.B. im Fall einer Regressionsanalyse mit einer Zielgröße und mehr als einer Einflussgröße, so kann wie folgt verallgemeinert werden:
Statt den zwei Vektoren X und Y betrachtet man nun eine Datenmatrix

$$X^\star = (x_{ij})^\star = \begin{pmatrix} x_{11} & \cdots & \cdots & \cdots & x_{1m} \\ \vdots & \ddots & & \star & \vdots \\ \star & & \ddots & & \vdots \\ \vdots & \star & & \ddots & \vdots \\ x_{n1} & \cdots & \cdots & \cdots & x_{nm} \end{pmatrix}$$

mit Spalten $j = 1, ..., m$, welche die Variablen darstellen, und Zeilen $i = 1, ..., n$, welche die Beobachtungen der Variablen darstellen. Diese Datenmatrix enthält als einen Vektor $x_{\cdot j}$ die Zielgröße und als restliche Vektoren die Einflussgrößen der Regressionsanalyse. Durch das Symbol $\star$ sind mögliche fehlende Werte dargestellt.
Die Bezeichnungen MCAR, MAR und NMAR können hier wie zuvor angegeben verwendet werden, statt Y müssen jedoch die beobachteten Komponenten der Datenmatrix $X^\star$ und statt X die fehlenden Komponenten der Datenmatrix $X^\star$ betrachtet werden.

Zudem werden bei fehlenden Daten zwei Muster ("patterns") unterschieden:

- Mit "unit nonresponse" wird der vollständige Ausfall einer Erhebungseinheit bezeichnet. Dazu kann es kommen, wenn Personen z.B. aufgrund von Verweigerung, Nicht-Erreichbarkeit oder aus anderen Gründen nicht auf eine Umfrage antworten.
- Von "item nonresponse" spricht man, wenn nur Werte bestimmter Variablen fehlen. Das kann passieren, wenn z.B. ein Befragter in einem Interview einzelne Antworten verweigert.

Annahmen über die Fehlendmechanismen in dieser Bachelorarbeit

In dieser Bachelorarbeit wurden nur Erhebungseinheiten mit "item nonresponse" betrachtet. Das heisst, bei den in dieser Arbeit betrachteten Probanden konnten in einzelnen Variablen Werte fehlen. War eine Erhebungseinheit mindestens in einer der drei Studien ISAAC II, SOLAR I oder SOLAR II vollständig ausgefallen ("unit nonresponse"), so liegen von dieser Einheit bezüglich der entsprechenden Studie keinerlei Daten vor. Diese Einheiten wurden in der vorliegenden Bachelorarbeit deshalb nicht betrachtet.

Die MAR-Annahme wird den meisten Imputationsalgorithmen zugrunde gelegt. In dieser Bachelorarbeit wurde ebenso davon ausgegangen, dass der Fehlendmechanismus MAR ist. Diese Annahme wurde getroffen, da es keinen Grund gab anzunehmen, dass die Wahrscheinlichkeit für das Fehlen eines Wertes in den potentiellen Confoundervariablen und den Tätigkeitsdaten von der Variable selbst abhängt. Dies wäre zum Beispiel in den Tätigkeitsdaten der Fall, wenn zum Beispiel die Wahrscheinlichkeit, dass eine Angabe fehlt, höher ist, wenn der Proband 30 Stunden pro Woche gearbeitet hat, als wenn er 15 Stunden pro Woche gearbeitet hat, was aber als sehr unlogisch erscheint.

3.2 Umgang mit fehlenden Daten

Liegt die Problematik fehlender Daten vor, so kann man sich darauf beschränken, nur die beobachteten Fälle zu betrachten, wie in Abschnitt 3.2.1 beschrieben wird. Alternativ kann man die fehlenden Werte auch mithilfe verschiedener Methoden ersetzen. Einige gängige Methoden hierfür werden in Abschnitt 3.2.2 vorgestellt.

3.2.1 Methoden für Betrachtung der beobachteten Werte

Complete Case Analysis

Bei dieser Methode werden nur die Fälle (Zeilen) betrachtet, bei denen für alle Variablen Werte vorliegen. Sobald bei einem Fall mindestens ein Wert in der Zielgröße oder in einer der Einflussgrößen fehlt, wird der entsprechende Fall komplett aus der Analyse ausgeschlossen.
Ein großer Nachteil dieser Methode ist ein potentieller Informationsverlust, der durch das Ausschließen aller unvollständigen Fälle entsteht. Dieser Informationsverlust kann insbesondere für Datensätze, die eine große Anzahl von Variablen oder eine große Anzahl von fehlenden Werten enthalten, erheblich sein. Fehlen nur sehr wenige Werte in einem Datensatz, so kann diese Methode zufriedenstellende Ergebnisse liefern. In epidemiologischen oder klinischen Studien entsteht durch das Ausschließen aller unvollständigen Fälle zudem ein ethisches Problem. Ein Proband, der an einer Studie teilnimmt, investiert viel Zeit in die Teilnahme und setzt sich eventuell den Risiken eines neuen Medikaments aus. Deswegen sollten seine Daten allein aus ethischer Sicht auf jeden Fall in die Analyse der Daten einbezogen werden, auch wenn diese unvollständig sein sollten.
Da man im Allgemeinen Aussagen über die gesamte Zielpopulation und nicht nur über die Teilpopulation der Probanden, die bei allen Fragen vollständige Angaben gemacht haben, machen will, erscheint die Complete Case Analyse eher als ungeeignet.
Zudem entsteht durch das Ausschließen der unvollständigen Fälle bei der Complete Case Analysis erstens ein Präzisionsverlust, zweitens kann es zu einer Verzerrung der Ergebnisse kommen, falls der Mechanismus, der den fehlenden Daten zugrunde liegt, nicht MCAR ist und die kompletten Fälle keine Zufallsstichprobe aus allen Fällen sind.
Problematisch ist insbesondere bei der Anwendung der Complete Case Analyse bei Regressionsanalysen, wenn die Wahrscheinlichkeit für das Fehlen der Werte von der Zielgröße abhängt. Die Complete Case Analyse liefert nur so lange eine erwartungstreue Schätzung wie die Wahrscheinlichkeit für das Fehlen der Kovariablenwerte nicht von der Zielgröße abhängt. Die Wahrscheinlichkeit für das Fehlen der Werte darf hierbei von den Kovariablenwerten selbst und auch von den fehlenden Werten abhängen, jedoch nicht von der Zielgröße [Toutenburg 2003].

Available Case Analysis

Bei dieser Methode werden alle Fälle betrachtet, die bei der jeweils betrachteten, also interessierenden Variable vollständig sind. Diese Methode nutzt alle beobachteten und somit verfügbaren Werte. Die gesamte zur Verfügung stehende Information wird also maximal ausgenutzt. Ein Nachteil dieser Methode ist, dass dadurch verschiedene Variablen unterschiedlich große Stichprobenumfänge haben. Dadurch kann es schon bei einfachen deskriptiven Statistiken zu Problemen der Vergleichbarkeit kommen, wenn die Daten nicht MCAR sind. Zudem sind bi- oder multivariate Modelle in diesem Fall nicht vergleichbar.

3.2.2 Imputationsmethoden - Ersetzen der fehlenden Werte

Um auch Fälle mit fehlenden Werten in die Datenanalyse mit einbeziehen zu können, werden häufig Methoden angewandt, durch welche fehlende Werte ersetzt werden. Beim Ersetzen fehlender Werte muss, egal welche Methode angewandt wird, immer mit einer Abweichung vom Original gerechnet werden, da die fehlenden Werte unbekannt sind [Toutenburg 2003]. Oft muss der Statistiker jedoch eine solche Abweichung und deren, in manchen Fällen gravierende, Auswirkungen in Kauf nehmen, da sonst die komplette Datenanalyse gefährdet wäre.
Innerhalb der Imputationsmethoden unterscheidet man zwischen der einfachen Imputation (Single Imputation), bei der genau ein Wert für jede fehlende Variable eingesetzt wird, und der multiplen Imputation, bei der mehr als ein Wert eingesetzt wird, um eine angemessene Schätzung der Unsicherheit, die mit der Imputation einhergeht, zu ermöglichen.

Mean Imputation

Bei dieser Methode werden die fehlenden Werte einer Variable durch das arithmetische Mittel der für diese Variable beobachteten Werte ersetzt. Manchmal kann es auch sinnvoll sein, innerhalb der beobachteten Daten einen klassen- oder gruppenspezifischen Mittelwert zu bilden, so dass zum Beispiel für Männer und Frauen verschiedene Mittelwerte gebildet und für die fehlenden Werte eingesetzt werden. Liegen kategoriale oder binäre Daten vor, so kann statt dem Mittelwert auch der Median bzw. Modus verwendet werden.
Problematisch ist bei der Mittelwertsimputation, dass die empirische Verteilung der Daten verzerrt wird. Dadurch können beispielsweise Varianzschätzer durch Standardmethoden nicht mehr konsistent geschätzt werden [Little 2002].

Regression Imputation

Hierbei werden fehlende Werte durch aus einer Regression vorhergesagte Werte ersetzt. Diese Regression ist eine Regression des fehlenden Eintrags auf die Einträge, die für die jeweilige Einheit beobachtet wurden. In einem Spezialfall, der "Stochastic Regression Imputation", wird zu den Einflüssgrößen zusätzlich noch ein Residuum addiert, welches die im vorhergesagten Wert enthaltene Unsicherheit ausdrücken soll.
Ein Vorteil dieser Methode ist, dass sie die Struktur innerhalb der Variablen ausnutzt und somit die Korrelationsstruktur der Variablen erhalten bleibt. Die Güte und Validität der Regression Imputation wird jedoch durch die Ursache des Fehlens (zufälliges Fehlen bzw. nichtzufälliges Fehlen) beeinflusst [Toutenburg und Heumann 2006].

Hot deck Imputation

Bei der Hot deck Imputation werden fehlende Werte ersetzt durch beobachtete Werte der betreffenden Variable, die aus "ähnlichen", vollständig beobachteten Einheiten gezogen werden. Die Ähnlichkeit wird dabei durch ein Abstandsmaß definiert, oft wird hierfür der euklidische Abstand verwendet.
Ein Vorteil der Hot deck Imputation ist, dass durch die imputierten Werte die empirische Verteilung der Daten nicht verzerrt wird, wie es zum Beispiel bei der Mittelwertsimputation der Fall ist. Jedoch gilt meist nur unter der im Allgemeinen unrealistischen Annahme MCAR, dass die Schätzer bei dieser Imputationsmethode unverzerrt sind [Little 2002].

Cold deck Imputation

Bei der Cold deck-Methode werden fehlende Werte ersetzt durch einen konstanten Wert aus einer externen Quelle, z.B. durch einen Erfahrungswert aus einer früheren Erhebung. Problematisch ist hierbei, dass eine geeignete Quelle für die Imputation gefunden werden muss, in der solch ein konstanter Wert angegeben wird. Die Qualität dieser Imputationsmethode hängt stark von der Wahl der Quelle ab, die zur Imputation verwendet wird.

Imputation durch Ziehen gemäß der Randverteilung der Daten

Bei dieser variablenbezogenen Methode werden fehlende Werte ersetzt durch Werte, die gemäß der Randverteilung der beobachteten Daten gezogen werden. Dabei wird jede Variable extra betrachtet. Ein Nachteil dieser Methode ist, dass sie die Abhängigkeitsstruktur in den Daten nicht berücksichtigt, da die Imputation der einzelnen Variablen unabhängig voneinander durchgeführt wird. Die Randverteilung aus den beobachteten

Daten wird nicht verändert, es kann jedoch die Korrelationsstruktur der Daten zerstört werden.

Problematik der Single Imputation

Bei der Single Imputation ist es so gut wie immer der Fall, dass die Unsicherheit bei der Imputation nicht berücksichtigt wird (außer bei der Regression Imputation, bei der ein Residuum addiert werden kann). Bei der Anwendung von Standard-Varianzformeln auf die vervollständigten Daten wird die Varianz der Schätzer deswegen systematisch unterschätzt. So erhöht sich zum Beispiel bei der Mean Imputation der Stichprobenumfang durch das Ersetzen der fehlenden Werte, nicht jedoch die Varianz. Deshalb werden Standardfehler systematisch unterschätzt [Little 2002]. Außerdem kann beispielsweise bei der Imputation durch Ziehen gemäß der Randverteilung der Daten die Korrelationsstruktur der Daten zerstört werden.

Multiple Imputation

Bei diesem Verfahren wird jeder fehlende Wert durch einen Vektor ersetzt, der m $\geq$ 2 Werte enthält. Es entstehen m vervollständigte Datensätze, indem man jeden fehlenden Wert zuerst durch den ersten im Vektor enthaltenen Wert ersetzt, woraus der erste vervollständigte Datensatz entsteht, dann durch den zweiten Wert um den zweiten vervollständigten Datensatz zu erhalten und so weiter. Die beobachteten Werte im Datensatz sind dabei fest, werden also nicht verändert, und nur die imputierten Werte unterscheiden sich.
Der Vorteil dieser Methode ist, dass jeder der m vervollständigten Datensätze anschließend mit einer beliebigen Standardmethode für vollständige Daten analysiert werden kann, ohne die Tatsache berücksichtigen zu müssen, dass die vollständigen Datensätze durch Imputation entstanden sind. Man berechnet also m Schätzungen aus den m Datensätzen, die anschließend zu einer endgültigen Schätzung kombiniert werden.
Gleichzeitig wird durch diese Methode auch die Stichprobenvariabilität berücksichtigt, die aufgrund der fehlenden Daten vorliegt. Die Variabilität der m imputierten Datensätze gibt die Unsicherheit, mit der die fehlenden Werte aufgrund der beobachteten Werte vorhergesagt werden, wider. Im Gegensatz zur Single Imputation existiert also bei der Multiplen Imputation das Problem der Varianzunterschätzung nicht mehr.
Ein weiterer Vorteil dieser Methode ist, dass man pro Datensatz nur einmal imputieren muss und die vervollständigten Datensätze für eine Reihe von Analysen verwenden kann, da bereits alle im Datensatz enthaltenen Variablen imputiert wurden. So muss nicht noch einmal neu imputiert werden, wenn eine neue Analyse durchgeführt wird.

Meist wird als einziger Nachteil dieser Imputationsmethode angeführt, dass im Gegensatz zur Single Imputation ein größerer Aufwand nötig ist, um die Imputation durchzuführen. In Zeiten von leistungsstarken Rechnern fällt diese Tatsache jedoch kaum noch ins Gewicht.

Die Idee der Multiplen Imputation wurde in den siebziger Jahren von Donald B. Rubin entwickelt.

Abbildung 3.1 verdeutlicht das Vorgehen bei der multiplen Imputation. Hier wurden für einen unvollständigen Datensatz drei vervollständigte Datensätze durch Multiple Imputation erstellt.

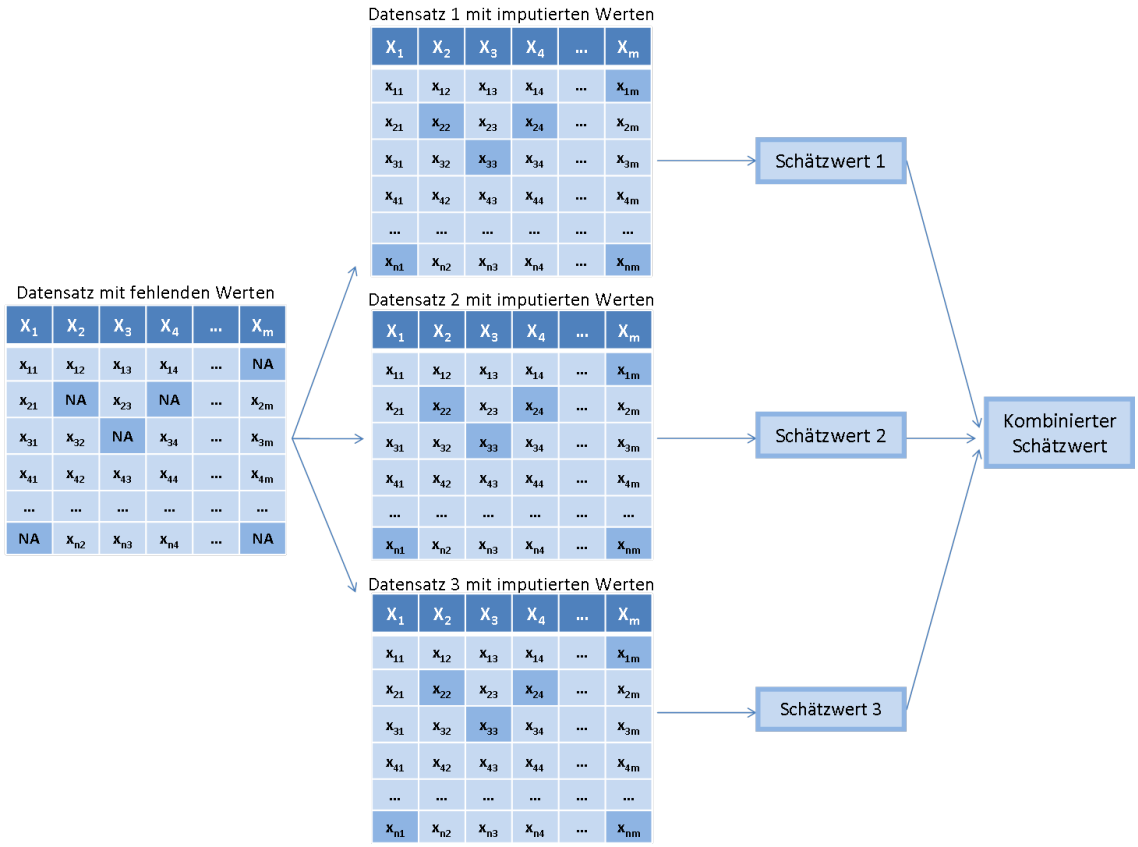


Abbildung 3.1: Multiple Imputation

Bemerkung

Neben den hier vorgestellten gibt es noch zahlreiche andere Imputationsmethoden. Es können auch Ideen verschiedener Imputationsmethoden zu einer Methode zusammengefasst und somit gemeinsam angewandt werden.

3.2.3 Kombination der Schätzer

Durch Multiple Imputation (MI) vervollständigte Datensätze können mit nahezu jeder Methode analysiert werden, die man bei vollständigen Datensätzen ohne fehlende Werte verwenden würde. Zum Beispiel können auf Basis der vervollständigten Datensätze lineare oder logistische Regressionsmodelle gerechnet werden. Ein Regressionsmodell muss dann m mal gefittet werden, also für jeden der vervollständigten Datensätze extra. Die Ergebnisse der Regressionsanalyse variieren dann je nach Datensatz, wodurch die Unsicherheit bei der Schätzung der fehlenden Werte widergespiegelt wird. Um insgesamt gültige Regressionskoeffizienten und die zugehörigen geschätzten Standardabweichungen zu erhalten, muss man die Koeffizientenschätzer, die man für die m imputierten Datensätze erhalten hat, kombinieren. Dazu gibt es folgende Regeln:
Sei $\widehat{Q}$ eine Schätzung des interessierenden Parameters und U eine Schätzung der Varianz des Parameterschätzers. $\widehat{Q}$ kann zum Beispiel eine Schätzung eines Regressionskoeffizienten sein und U der zugehörige Schätzer für die Varianz von $\widehat{Q}$. Aus der Analyse der m vervollständigten Datensätze erhält man somit m gleichermaßen plausible Schätzer $\widehat{Q_1}$, $\widehat{Q_2}$,..., $\widehat{Q_m}$ und deren zugehörige Varianzen U_1,U_2,...,U_m. Der MI-Schätzer ist dann gegeben durch:

$$\overline{Q} = \frac{1}{m}\sum_{i=1}^{m}\widehat{Q_i} \tag{3.1}$$

Die Varianz des Schätzers besteht aus zwei Komponenten: aus der Varianz innerhalb jedes vervollständigten Datensatzes und der Varianz zwischen den vervollständigten Datensätzen. Die Varianz innerhalb jedes Datensatzes ist das arithmetische Mittel der geschätzten Varianzen:

$$\overline{U} = \frac{1}{m}\sum_{i=1}^{m}U_i \tag{3.2}$$

Die Varianz zwischen den Datensätzen ist die Stichprobenvarianz der Schätzer selbst:

$$B = \frac{1}{m-1}\sum_{i=1}^{m}(\widehat{Q_i} - \overline{Q})^2 \tag{3.3}$$

Die Gesamtvarianz T entspricht der Summe der beiden Komponenten mit einem zusätzlichen Korrekturfaktor für den Simulationsfehler in $\overline{Q}$:

$$T = \overline{U} + \left(1 + \frac{1}{m}\right)B \tag{3.4}$$

Die Wurzel aus T ist dann die zum Schätzer $\overline{Q}$ gehörige Standardabweichung.
Ein grobes 95%-Konfidenzintervall kann berechnet werden durch die Formel:

$$\overline{Q} \pm 2\sqrt{T} \tag{3.5}$$

Meist ist es jedoch besser Konfidentintervalle durch die Approximation

$$\overline{Q} \pm t_{df}\sqrt{T} \tag{3.6}$$

zu berechnen, wobei t_{df} für das Quantil der Student´s t-Verteilung mit folgenden Freiheitsgraden steht:

$$df = (m-1)\left(1 + \frac{m\overline{U}}{(m+1)B}\right)^2 \tag{3.7}$$

[Schafer und Olsen 1998].

KAPITEL 4

Datenmanagement

4.1 Datengrundlage

Für diese Bachelorarbeit wurden nur die Daten derjenigen Probanden verwendet, die an allen drei Studien (ISACC II, SOLAR und SOLAR II) teilgenommen hatten und deren Daten aus SOLAR II zum Zeitpunkt des Beginns dieser Arbeit bereits mit den Daten aus ISAAC II und SOLAR verknüpft werden konnten. Die Probanden mussten also ihre Einverständnis zur longitudinalen Verknüpfung ihrer Daten gegeben haben. Außerdem mussten die beruflichen Tätigkeiten der Probanden zum Zeitpunkt des Beginns dieser Arbeit schon vollständig kodiert worden sein.
Bei den Probanden, deren berufliche Tätigkeiten zu diesem Zeitpunkt noch nicht vollständig kodiert waren, kann davon ausgegangen werden, dass diese zufällig fehlten und durch das Ausschließen dieser Probanden aus der Analyse somit keine Verzerrung zu erwarten ist.
Bei allen vorliegenden Probanden wurde zudem überprüft, ob in einer der drei Studien Angaben zu Neurodermitis, allergischer Rhinitis oder Asthma fehlten. Fehlte mindestens eine solche Angabe, so wurde der entsprechende Proband aus der Analyse ausgeschlossen, da diese medizinischen Daten nicht imputiert werden durften. Diese Probanden wurden aus der Analyse ausgeschlossen, da konservativ vorgegangen wurde, d.h., nur diejenigen Probanden, bei denen klar war, ob sie an einer der zuvor genannten Erkrankungen zu einem der drei Beobachtungszeiten litten oder nicht, sollten betrachtet werden. Es sollten also nur "sichere" Krankheitsfälle in der Analyse betrachtet werden.
Aufgrund der oben genannten Bedingungen konnten von den 1.966 Probanden mit vorliegenden Daten für alle drei Studien, deren Daten vom Institut und der Poliklinik für Arbeits-, Sozial- und Umweltmedizin der Ludwig-Maximilians-Universität München zur Verfügung gestellt wurden, 1.187 Probanden für die in dieser Arbeit durchgeführten Analysen verwendet werden. Abbildung 4.1 gibt eine Übersicht über die Datengrundlage für die vorliegende Bachelorarbeit.

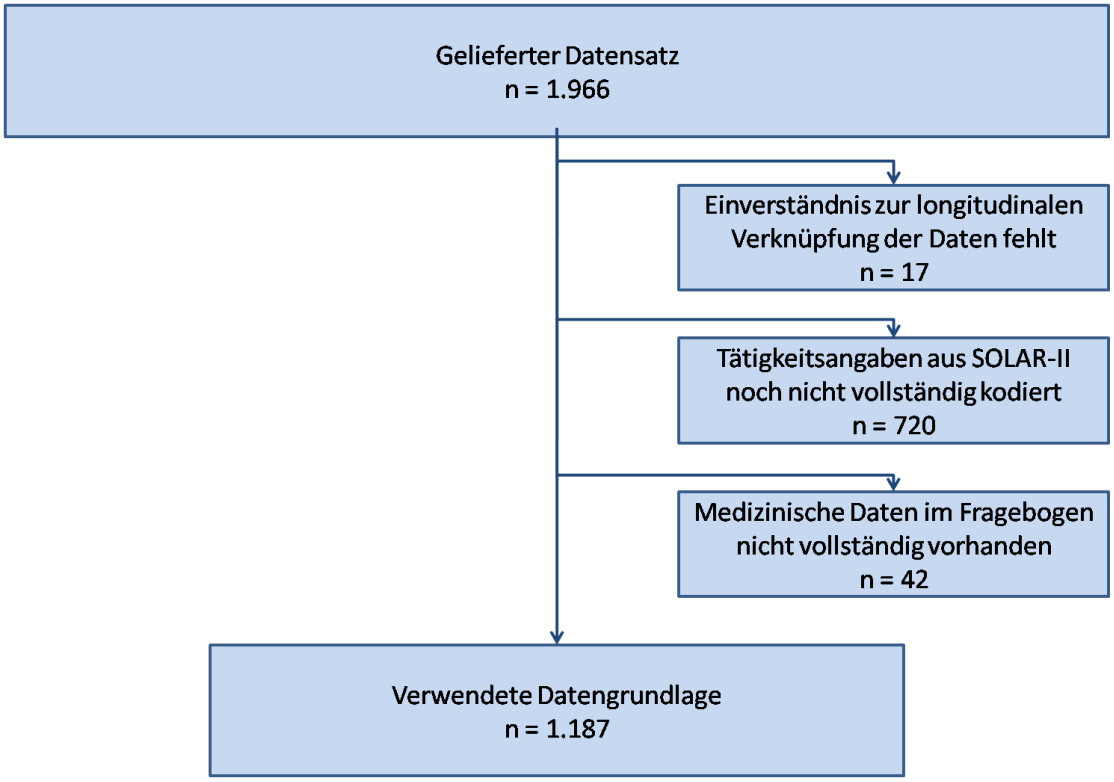


Abbildung 4.1: Datengrundlage der Analysen

4.2 Datenbereinigung

An den Datensatz, der nur noch die relevanten Fälle enthält, wurden noch einige nachträglich gelieferte Zusatzinformationen, die für die weiteren Analysen erforderlich waren, angefügt. Es mussten zudem noch zahlreiche Kodierungen vorgenommen werden, um die für die Analysen relevanten Variablen in der benötigten Kodierung vorliegen zu haben. Die Variablen für Asthma und allergische Rhinitis in SOLAR II mussten aus mehreren Variablen gebildet werden, analog zur Kodierung in SOLAR I. Die Angaben zur Berufssituation in SOLAR I und SOLAR II wurden in kategoriale Variablen umkodiert und Doppelnennungen korrigiert, so dass für jeden Probanden, wie gewünscht, pro Studie nur eine aktuelle Berufssituation vorlag. Alle binären Variablen aus den drei Studien wurden einheitlich 0/1-kodiert. Die ausführliche Beschreibung aller vorgenommenen Kodierungen ist dem Anhang (Kapitel A) zu entnehmen.

4.2.1 Korrekturen der Tätigkeitsdaten

Die Tätigkeitsdaten aus SOLAR I und SOLAR II lagen in zwei verschiedenen Datensätzen vor. Bevor jedoch mit diesen Tätigkeitsdaten gearbeitet werden konnte, mussten die im Folgenden beschriebenen Korrekturen an den Daten vorgenommen werden.

Zunächst wurden alle relevanten Tätigkeitsangaben auf Plausibilität geprüft. Jeder Proband, der mindestens einen Eintrag zu den Tätigkeitsangaben (Frage 66 in SOLAR I bzw. Frage 93 in SOLAR II) gemacht hatte, musste auch die vorhergehende Frage (Frage 65 in SOLAR I: "Haben Sie schon einmal irgendeine Arbeit / irgendeinen Ferienjob gehabt?" bzw. Frage 92 in SOLAR II: "Haben Sie seit der letzten SOLAR-Studie (2003/2004) irgendeine Arbeit / irgendeinen Ferienjob für mindestens 1 Monat gehabt?") mit "ja" beantwortet haben. Bei den Probanden, bei denen dies nicht der Fall war, wurde die Angabe zu Frage 65 bzw. Frage 92 entsprechend korrigiert.

Ebenso wurden die Angaben zu Beginn und Ende der Tätigkeit auf Plausibilität geprüft. Falls das Ende der Tätigkeit vor dem Anfang der Tätigkeit lag, so wurde nochmal anhand der Originalfragebögen der Probanden überprüft, ob die Angaben richtig in die Datenbank eingegeben wurden. Handelte es sich um einen Eingabefehler, so wurden Beginn und Ende der Tätigkeit entsprechend korrigiert. Wurden die Angaben tatsächlich so gemacht, wurden Anfang und Ende auf fehlend (NA) gesetzt.
Bei den Tätigkeiten, bei denen nur die Angabe zum Ende der Tätigkeit fehlte, alle anderen Angaben (ISCO-Code, Anfangsmonat und Anfangsjahr der Tätigkeit, Anzahl der Wochenstunden) jedoch vollständig vorlagen, wurde davon ausgegangen, dass die Tätigkeit zum Zeitpunkt der Befragung noch ausgeführt wurde. Aus diesem Grund wurde für das Ende ein Ersatzende eingesetzt und die entsprechende Tätigkeit als vollständig angesehen. Als Ersatzende wurde das Ausfülldatum des Fragebogens verwendet. War das Ausfülldatum fehlend, so wurde das Einscanndatum des Fragebogens verwendet. Das Ersatzende wurde in insgesamt 469 Zeilen (169 Zeilen aus SOLAR I und 300 Zeilen aus SOLAR II) eingesetzt.
Nachdem das Ersatzende in den oben genannten Fällen eingesetzt wurde, musste nochmals überprüft werden, ob neue Fälle entstanden waren, bei denen das Ersatzende der Tätigkeit vor dem Anfang der Tätigkeit lag. In insgesamt sieben Zeilen (zwei Zeilen aus SOLAR I und fünf Zeilen aus SOLAR II) trat der Fall auf, dass der Anfangszeitpunkt der Tätigkeit nun nach dem Ersatzende der Tätigkeit, d.h. nach dem Ausfüllzeitpunkt bzw. Einscanndatum des Fragebogens lag. Da dies bedeutet, dass im entsprechenden Fall eine Tätigkeit angegeben wurde, die erst nach dem Ausfüllzeitpunkt bzw. Einscanndatum des Fragebogens begonnen wurde und deshalb zum Zeitpunkt der Teilnahme an der Studie noch keine Exposition vorlag, wurde bei diesen Tätigkeitsangaben der ISCO-Code auf 97 (zukünftige Tätigkeit) und die Exposition auf 0 (nicht exponiert) gesetzt.

Bei Probanden, die zwar eine Tätigkeit angegeben hatten, jedoch nicht klar war, um welche Tätigkeit es sich genau handelte und deswegen kein passender ISCO-Code gefunden werden konnte, wurde der ISCO-Code auf 94 gesetzt und die Expositionen auf 0

(nicht exponiert). Bei diesen Personen wurde somit angenommen, dass keine Exposition vorlag. In SOLAR II gab es zusätzlich noch den Code 98 für Schüler und Studenten und in SOLAR I den Code 95 für Hausfrauen. Hier wurde genauso vorgegangen und die Expositionen auf 0 gesetzt.

Es wurde zudem überprüft, ob der ISCO-Code bei allen Tätigkeiten der Probanden vorlag. Fehlende Werte durfte es hier nicht geben. Der ISCO-Code sollte außerdem nur den Wert 9999 (SOLAR I) bzw. 8888 (SOLAR II) haben, wenn keine Tätigkeit vorlag. In SOLAR II waren jedoch auch Fälle enthalten, die trotz Tätigkeitsangaben mit dem Code 8888 kodiert wurden. Nach Absprache mit Frau Kellberger wurden die zugehörigen Probanden, soweit erreichbar, kontaktiert, woraufhin eine Korrektur der ISCO-Codes und der zugehörigen Expositionen vorgenommen werden konnte. In SOLAR I war es ebenso der Fall, dass zwei Tätigkeiten mit dem ISCO-Code 9999 kodiert wurden, obwohl Tätigkeitsangaben vorhanden waren. In einem der beiden Fälle wurden zwar Tätigkeitsangaben gemacht, es fehlte jedoch die Angabe, welche Tätigkeit ausgeübt wurde. Hier wurde der ISCO-Code auf 94 gesetzt.

Die Wochenstundenangaben wurden auf Plausibilität geprüft. Einzelne Probanden gaben an, mehr als 60 Wochenstunden gearbeitet zu haben. Da diese Angaben in Absprache mit Frau Prof. Dr. Radon als unplausibel eingestuft wurden, wurden bei diesen Probanden die Wochenstunden auf fehlend (NA) gesetzt. Die Tätigkeitsangaben, bei denen weniger als 12 Wochenstunden angegeben wurden, wurden überprüft, ob es wahrscheinlich ist, dass Stunden pro Tag statt Stunden pro Woche angegeben wurden. Bei allen Probanden die als Tätigkeit Ausbildung, Zivildienst, Bundeswehr, Praktikum oder Freiwilliges Soziales Jahr und weniger als 12 Wochenstunden angegeben hatten, wurde davon ausgegangen, dass sie Stunden pro Tag statt Stunden pro Woche angegeben hatten. Deshalb wurde die Anzahl der Wochenstunden in diesen Fällen mit fünf multipliziert, um Stunden pro Woche zu erhalten. Bei denselben Probanden wurden anschließend auch alle weiteren Tätigkeitsangaben (der entsprechenden Studie) überprüft, da davon ausgegangen werden musste, dass sie auch bei ihren restlichen Tätigkeitsangaben denselben Fehler gemacht haben könnten. War dies der Fall, so wurden die Wochenstundenangaben analog korrigiert.

Ein Proband hatte in SOLAR II statt der vorgesehenen fünf Tätigkeitsangaben Angaben zu sechs Tätigkeiten gemacht. Da es sich aber bei zwei aufeinander folgenden Angaben um die gleiche Tätigkeit, und somit um den gleichen ISCO-Code, handelte und das Ende der einen Tätigkeit gleichzeitig der Beginn der nächsten Tätigkeit war, konnten diese beiden Angaben zu einer Tätigkeit mit entsprechend längerer Dauer zusammenge-

fasst werden. Somit lagen bei allen Probanden maximal fünf Tätigkeitsangaben vor.

Waren in der Job-Exposure-Matrix fehlende Werte (SOLAR II) bzw. die Ziffer 9 (SOLAR I) enthalten (bei den Probanden, die nicht gearbeitet hatten), so wurden diese Einträge auf 0 gesetzt, da die JEM nur die Werte 1 (exponiert) und 0 (nicht exponiert) enthalten sollte und die entsprechenden Probanden offensichtlich nicht exponiert waren.

Tabelle 4.1 gibt eine Übersicht über die Korrekturen, die an den Tätigkeitsdaten vorgenommen wurden.

Korrekturen	**Anzahl Fälle**
Tätigkeitsangaben vorhanden, aber Frage ob gearbeitet wurde verneint	37 Fälle
Ende der Tätigkeit lag vor Anfang der Tätigkeit	5 Fälle
Nur Ende der Tätigkeit fehlte (Ersatzende wurde eingesetzt)	469 Fälle
Mehr als 60 Wochenstunden angegeben	9 Fälle
Verwechslung Stunden pro Woche mit Stunden pro Tag	29 Fälle
Tätigkeitsangaben trotz ISCO-Code 8888 bzw. 9999 vorhanden	5 Fälle
Sechs statt fünf vorgesehenen Tätigkeiten angegeben	1 Fall

Tabelle 4.1: Korrekturen der Tätigkeitsdaten

Tabelle 4.2 enthält die in dieser Arbeit zusätzlich eingeführten "ISCO-Codes" zur Berufskodierung. Hier sind auch die Fälle aus den Korrekturen der Tätigkeiten enthalten, die eine Änderung des ISCO-Codes in einen der ISCO-Codes 94,95,97 oder 98 nach sich zogen. Bei den Probanden mit ISCO-Code 94,95,97 oder 98 wurde konservativ vorgegangen und die Expositionen auf 0 (nicht exponiert) gesetzt, da es besser ist, eine Unterschätzung der Exposition zu erhalten also eine Überschätzung.

ISCO-Code	**Beschreibung**	**Anzahl der Probanden**
94	Tätigkeit nicht codierbar	15
95	Keine berufliche Tätigkeit (Hausfrau)	1
98	Keine berufliche Tätigkeit (Schüler/Student)	6
97	Zukünftige Tätigkeit	7

Tabelle 4.2: Übersicht über die zusätzlich eingeführten "ISCO-Codes"

Durch die JEM von S. Kennedy kann zwar jeder Tätigkeit eine bestimmte Exposition zugeordnet werden, ob der einzelne Proband aber auch wirklich dieser Exposition ausgesetzt war, kann nicht beurteilt werden, da jeder Arbeitsplatz individuell ist. So kann es bei einer Person mit der Tätigkeit Bäcker beispielsweise sein, dass sie wirklich am Backen der Backwaren oder an der Teigzubereitung beteiligt ist und dadurch während des ganzen Arbeitstages gegenüber Mehlstaub exponiert ist. Eine andere Person mit der Tätigkeit Bäcker, die zum Beispiel ausschließlich in der Zulieferung von Backwaren oder im Verkauf der Backwaren tätig ist, ist im Gegensatz dazu wahrscheinlich nur geringfügig oder gar nicht gegenüber Mehlstaub exponiert. Falls die Angaben, die zu den Tätigkeiten gemacht wurden, recht konkret sind, so kann die Exposition gut mit Hilfe der JEM abgeschätzt werden. Falls man aber nicht sicher sein kann, was der entsprechende Proband tatsächlich getan hat, ist es besser, konservativ vorzugehen und eine bestimmte Exposition eher nicht anzunehmen, weil es in diesem Fall noch unsicherer ist, ob die Expositionszuordnungen aus der JEM tatsächlich zutreffen.

4.3 Auswahl der Probanden mit vollständigen Tätigkeitsangaben

Um die Probanden mit vollständigen Tätigkeitsangaben in SOLAR I bzw. SOLAR II herauszufinden, mussten zunächst einige neue Variablen gebildet werden. Zuerst wurde pro Studie eine Variable gebildet, die angibt, ob in der entsprechenden Zeile überhaupt Tätigkeitsangaben vorhanden sind. Diese Tätigkeitsangaben mussten dabei jedoch nicht vollständig sein, sie konnten auch unvollständig sein. Darauf basierend wurde anschließend eine Variable gebildet, die für jeden Probanden die Gesamtanzahl der Tätigkeitsangaben enthält, die also angibt, wie viele Tätigkeitsangaben dieser Proband pro Studie insgesamt gemacht hat.
In SOLAR II konnte bei den Tätigkeitsangaben zusätzlich von den Probanden angegeben werden, dass nur Tätigkeiten mit weniger als acht Wochenstunden ausgeführt wurden. Wurde dies angegeben, so wurde auch diese Zeile wie ein Eintrag mit Tätigkeitsdaten behandelt.
Eine zusätzliche Variable gibt an, ob die entsprechende Zeile vollständig ausgefüllt wurde oder ob in dieser Zeile Angaben fehlen.

In den folgenden Fällen wurde eine Zeile als vollständig kodiert:

- Die Tätigkeitsangaben in dieser Zeile waren vollständig ausgefüllt, d.h. Angaben zu Beginn und Ende der Tätigkeit, Wochenstunden und ISCO-Code lagen vor.
- Der ISCO-Code 94, 95, 97 oder 98 lag vor. Da in diesen Fällen die Exposition auf 0 gesetzt wurde, werden diese Fälle zu den vollständigen Zeilen gezählt, auch wenn Angaben fehlten.
- Es wurde angegeben, dass gearbeitet wurde, jedoch keine Tätigkeitsangaben gemacht oder die Angabe, ob gearbeitet wurde, fehlte und es wurden keine Tätigkeitsangaben gemacht. In diesen Fällen wurde konservativ vorgegangen und die Exposition auf 0 gesetzt.
- Es wurde weniger als acht Wochenstunden gearbeitet. Auch hier wurde die Exposition auf 0 gesetzt und die Zeilen zählen zu den vollständigen Zeilen, auch wenn Angaben fehlten.
- Es wurde angegeben, dass ausschließlich Tätigkeiten mit weniger als acht Wochenstunden ausgeführt wurden (nur in SOLAR II möglich). Hier lag keine Exposition vor und die enstprechenden Zeilen gelten als vollständig.
- Da bei Zeilen, bei denen nur der Endzeitpunkt der Tätigkeit fehlte, die Tätigkeitsangaben jedoch sonst vollständig waren, dieser durch ein Ersatzende ersetzt wurde, zählen auch diese Zeilen zu den vollständigen.

Darauf basierend konnte eine Variable gebildet werden, die für jeden Probanden die Anzahl der vollständig ausgefüllten Zeilen in SOLAR I bzw. SOLAR II enthält.
Mit Hilfe der beiden Variablen, die die Anzahl der vollständig ausgefüllten Zeilen pro Studie und die Anzahl der Tätigkeitsangaben pro Studie enthalten, konnte schließlich eine Variable gebildet werden, die angibt, ob alle fünf möglichen Tätigkeitsangaben aus der entsprechenden Studie vollständig sind, der Proband also in dieser Studie zu den Probanden mit vollständigen Tätigkeitsangaben zählt.

Ein Proband gilt pro Studie als Proband mit vollständigen Tätigkeitsangaben, wenn einer der folgende Fälle zutrifft:

- **Der Proband hat in dieser Studie nicht gearbeitet**
  Hat ein Proband angegeben, nicht gearbeitet zu haben und auch keine Tätigkeitsangaben gemacht, so gilt er als vollständig.

- **Der Proband hat in dieser Studie vollständige Tätigkeitsangaben gemacht**
  Hat ein Proband in SOLAR I bzw. SOLAR II gearbeitet, mindestens eine Tätigkeitsangabe gemacht und die Anzahl der ausgefüllten Zeilen (Tätigkeitsangaben) ist gleich der Anzahl der vollständig ausgefüllten Zeilen (Tätigkeitsangaben), so gilt er als vollständig. Auch die Probanden, die in SOLAR II angegeben hatten, ausschließlich Jobs mit weniger als acht Wochenstunden ausgeübt zu haben, gelten als vollständig.

- **Der Proband hat angegeben, während dieser Studie gearbeitet zu haben und keine Tätigkeitsangaben gemacht oder der Proband hat keine Angabe gemacht, ob er gearbeitet hat und keine Tätigkeitsangaben gemacht**
  Hat ein Proband angegeben, gearbeitet zu haben, jedoch anschließend keine Tätigkeitsangaben gemacht oder die Angabe, ob gearbeitet wurde fehlt und es wurden keine Tätigkeitsangaben gemacht, so wurde nach Absprache mit Frau Prof. Dr. Radon konservativ vorgegangen, d.h. es wurde angenommen, dass keine Exposition vorliegt. Deshalb gilt er als vollständig.

Ein Proband gilt pro Studie als Proband mit unvollständigen Tätigkeitsangaben, wenn der folgende Fall zutrifft:

- **Der Proband hat in dieser Studie lückenhafte Tätigkeitsangaben gemacht**
  War bei einem Probanden die Anzahl der ausgefüllten Zeilen (Tätigkeitsangaben) nicht gleich der Anzahl der vollständig ausgefüllten Zeilen (Tätigkeitsangaben), so gilt er als unvollständig.

In SOLAR I haben 1.144 Probanden und in SOLAR II 1.135 Probanden vollständige Tätigkeitsangaben gemacht. Die restlichen 43 Probanden in SOLAR I und 52 Probanden in SOALR II haben unvollständige Tätigkeitsangaben gemacht.

Tabelle 4.3 gibt getrennt für SOLAR I und SOLAR II eine Übersicht über die Probanden, die vollständige Tätigkeitsangaben gemacht haben.

Probanden	Anzahl Personen in SOLAR	Anzahl Personen in SOLAR II
die nicht gearbeitet haben (keine Exposition vorhanden)	477	500
mit durchgängig vollständigen Berufsangaben (Exposition kann vorhanden sein)	342	630
mit unklarer Arbeitssituation od.komplett fehlenden Tätigkeitsangaben (Exposition als nicht vorhanden angenommen)	325	5
mit vollständigen Tätigkeitsangaben	1.144	1.135

Tabelle 4.3: Übersicht über die Probanden mit vollständigen Tätigkeitsangaben in SOLAR I bzw. SOLAR II

Zusammenführen der Tätigkeitsdaten aus SOLAR I und II

Die Tätigkeitsdaten aus SOLAR I lagen als Datensatz vor, in dem jeder Proband genau eine Zeile hatte, in der bis zu fünf Tätigkeitsangaben enthalten waren. Für spätere Berechnungsschritte war es nötig, die Tätigkeitsdaten so zeilenweise anzuordnen, dass pro Tätigkeit eine Zeile vorhanden ist, d.h., pro Proband lagen nun fünf Zeilen für bis zu fünf Tätigkeitsangaben vor.
Die Tätigkeitsdaten aus SOLAR II lagen in einem separaten Datensatz vor, der die Tätigkeitsdaten bereits in zeilenweiser Form enthielt, d.h., pro Proband waren bereits fünf Zeilen (für bis zu fünf Tätigkeitsangaben) enthalten. Aus dieser Datei wurden nur diejenigen Probanden ausgewählt, für die bereits alle fünf Tätigkeitsangaben vollständig kodiert vorlagen. Nur diese Probanden wurden für die Analysen verwendet, da bei den restlichen, noch unvollständig kodierten Probanden, mögliche Tätigkeiten und somit auch die zugehörigen Expositionen nicht berücksichtigt werden würden.
Aus den beiden separaten Datensätzen mit Tätigkeitsangaben aus SOLAR I bzw. SOLAR II wurde ein Datensatz erstellt, der nun die Tätigkeitsangaben aus beiden Studien enthält. In dieser Datei sind also pro Proband zehn Zeilen für bis zu zehn Tätigkeitsangaben vorhanden. Damit bei jedem Probanden die erste ausgeführte Tätigkeit in der ersten der zehn Zeilen steht, wurde die Datei nach knr (Identifikationsnummer des Probanden), Anfangsjahr und Anfangsmonat sortiert.
In dieser Datei für die Tätigkeitsdaten aus SOLAR I und SOLAR II wurden nun einige Hilfsvariablen erstellt, die für spätere Analysen der Tätigkeitsdaten nötig sind.

Zunächst wurden folgende Variablen gebildet, die die separat vorliegenden Informationen aus SOLAR I und SOLAR II zusammenführen:

- Anzahl der Einträge:
  Gibt pro Proband die Gesamtanzahl der Tätigkeitsangaben aus SOLAR I und SOLAR II an
- Anzahl der vollständigen Zeilen:
  Gibt pro Proband die Gesamtanzahl der vollständigen Zeilen aus SOLAR I und SOLAR II an

Anschließend wurden noch einige weitere Hilfsvariablen gebildet. Da nur bei Tätigkeiten, bei denen mindestens acht Wochenstunden gearbeitet wurde, auch die Exposition betrachtet wurde, war eine Indikatorvariable nötig, die pro Tätigkeit angibt, ob die jeweilige Tätigkeit für mindestens acht Stunden pro Woche ausgeführt wurde. Um bei jedem Probanden feststellen zu können, wie viele Tätigkeiten mit mindestens acht Wochenstunden er ausgeführt hat, wurde eine Variable gebildet, die pro Proband angibt, wie viele Tätigkeiten mit mindestens acht Wochenstunden er ausgeübt hat.
Als Einflussgröße für die Regressionsmodelle wird eine Variable benötigt, die angibt, ob jemals (in SOLAR I oder SOLAR II) gearbeitet wurde. Diese Variable hat nur dann den Wert 1 ("ja"), wenn der jeweilige Proband mindestens eine Tätigkeit für mindestens acht Stunden pro Woche ausgeübt hat.

Pro Tätigkeitsangabe wurde außerdem aus den Angaben zu Anfang und Ende der Tätigkeit die Dauer der Tätigkeit in Monaten berechnet. Fehlte mindestens eine Angabe zu Anfang oder Ende der Tätigkeit (Anfangsmonat, Anfangsjahr, Endmonat, Endjahr), so konnte die Dauer nicht berechnet werden und wurde deswegen auf fehlend (NA) gesetzt. Bei den Probanden mit ISCO-Code 94,95,97 oder 98 wurde die Dauer auf 0 gesetzt, da diese Probanden als Probanden mit vollständigen Tätigkeitsangaben gesehen wurden. Da bei diesen Probanden die Expositionen auf 0 ("nicht exponiert") gesetzt wurden, konnte auch die Dauer auf 0 gesetzt werden, da bei den Expositionsberechnungen so eh resultieren wird, dass sie keiner Exposition ausgesetzt waren.

Auswahl der Probanden mit insgesamt vollständigen Tätigkeitsdaten

Ein Proband gilt als insgesamt vollständig, wenn er in beiden Studien (SOLAR I und SOLAR II) als vollständig definiert wurde. Von den ursprünglich 1.187 Probanden konnten gemäß obiger Definition 1.094 Probanden (92%) mit vollständigen Tätigkeitsdaten ausgewählt werden.

Probanden mit insgesamt vollständigen Tätigkeitsangaben
1094 Probanden, d.h. 92% der insgesamt 1187 Probanden haben insgesamt vollständige Tätigkeitsangaben gemacht.

In dieser Arbeit werden auf Basis der Probanden mit insgesamt vollständigen Tätigkeitsangaben logistische Regressionsmodelle gerechnet.

Auswahl der Probanden mit insgesamt unvollständigen Tätigkeitsdaten

In dieser Arbeit wird eine Simulation durchgeführt, bei der fehlende Daten im Datensatz, der nur die Probanden mit vollständigen Tätigkeitsangaben enthält, nach bestimmten Fehlendmustern (auf Basis des Gesamtdatensatzes) erzeugt werden. Als Grundlage für diese Simulation musste allerdings zunächst analysiert werden, welche Fehlendmuster im Gesamtdatensatz, der die Tätigkeitsangaben aller 1.187 Probanden enthält, auftreten. Tabelle 4.4 soll das vorliegende Fehlendmuster detailliert darstellen.

Fehlendmuster	**Anzahl Zeilen**
alle Angaben fehlen bis auf ISCO-Code	13 Zeilen
Wochenstunden fehlen	27 Zeilen
Wochenstunden, Zeitangaben zum Ende der Tätigkeit fehlen	14 Zeilen
Zeitangaben zum Anfang und Ende der Tätigkeit fehlen	14 Zeilen
Anfangsmonat und Endmonat fehlen	17 Zeilen
Zeitangaben fehlen (bis auf Anfangsjahr)	15 Zeilen
Anfangsjahr und Endjahr fehlen	1 Zeile
Anfangsmonat fehlt	1 Zeile
Wochenstunden und Zeitangaben (bis auf Anfangsjahr) fehlen	3 Zeilen
Anfangsmonat und Wochenstunden fehlen	1 Zeile
Zeitangaben (bis auf Anfangsmonat) fehlen	1 Zeile
Endjahr fehlt	1 Zeile

Tabelle 4.4: Übersicht über das vorliegende Fehlendmuster

KAPITEL 5

Imputation der fehlenden Werte in den potentiellen Confoundervariablen

Zunächst wurden nur die potentiellen Confoundervariablen (getrennt von den Tätigkeitsdaten) betrachtet und fehlende Werte in diesen Variablen wurden imputiert. Von den 1.187 Probanden, deren Daten für diese Bachelorarbeit vorlagen, haben 1.050 Probanden (88%) vollständige Angaben in den potentiellen Confoundervariablen gemacht. Bei den restlichen 137 Probanden fehlte mindestens eine Angabe in den potentiellen Confoundervariablen.
Die Tabellen 5.1 bis 5.4 geben eine Übersicht über die fehlenden Werte in den einzelnen Variablen. Zunächst werden in Tabelle 5.1 die Variablen aus ISAAC II aufgeführt. Anschließend wird in den Tabellen 5.2 und 5.3 die Struktur der Variablen, die aus SOLAR I und SOLAR II stammen, dargestellt. In Tabelle 5.4 werden die Variablen, die als Zusatzinformation für die Imputation verwendet werden, dargestellt. Wir befinden uns hier in der glücklichen Situation, dass für die Imputation mehr Variablen zur Verfügung stehen als für die eigentliche Datenanalyse. Dies ist generell eher selten der Fall.
In den Variablen "Studienzentrum" (aus ISAAC II) und "Geschlecht" (aus SOLAR II) waren keine fehlenden Werte vorhanden. Daher mussten anschließend auch keine Werte imputiert werden. Bei den medizinischen Variablen für Neurodermitis, allergische Rhinitis und Asthma zum Zeitpunkt ISAAC II bzw. SOLAR I musste auch nicht imputiert werden, weil Probanden mit fehlenden Werten in diesen Variablen aus der Analyse in dieser Bachelorarbeit ausgeschlossen wurden. Auch bei den medizinischen Variablen für allergische Rhinitis und Asthma zum Zeitpunkt SOLAR II musste nichts imputiert werden, weil dadurch, dass nur Probanden betrachtet wurden, die hier Angaben gemacht hatten, keine fehlenden Werte in diesen Variablen vorlagen. Diese Variablen werden der Vollständigkeit halber trotzdem in den Tabellen mit aufgeführt.

In dieser Bachelorarbeit wurden mittels der Methode der Imputation durch Ziehen gemäß der Randverteilung der Daten, die bereits in Kapitel 3 kurz vorgestellt wurde, zwei vervollständigte Datensätze aus dem unvollständigen Datensatz, der die Confoundervariablen aus ISAAC, SOLAR I und SOLAR II enthält, erstellt. Die Variablen "Jemals gearbeitet" (ja/nein) und die Job-exposure-Variablen gingen an dieser Stelle also nicht mit ein.
Durch multiple Imputation mit dem R-Paket AMELIA II wurden zudem nochmals drei vervollständigte Datensätze auf der Basis des gleichen unvollständigen Datensatzes erstellt. Beide Methoden werden im Folgenden ausführlich beschrieben. Abbildung 5.1 soll zuvor das Vorgehen bei der Imputation der fehlenden Werte in den Confoundervariablen veranschaulichen.

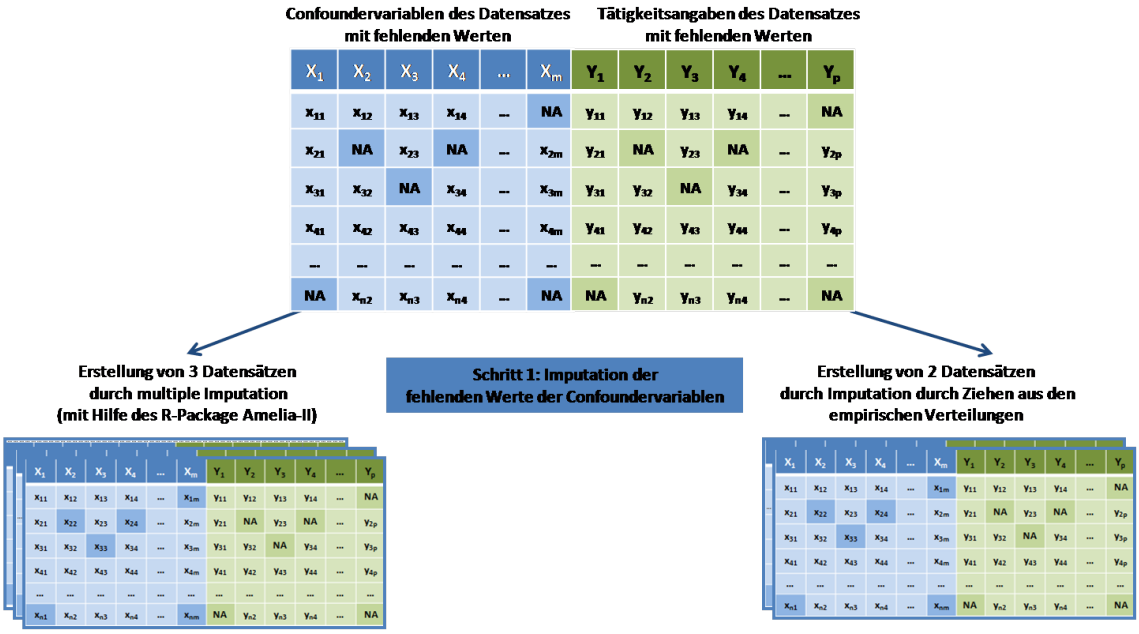


Abbildung 5.1: Imputation der fehlenden Werte in den potentiellen Confoundervariablen

Variablenbeschreibung und Häufigkeit	**vorhandene Werte**	**fehlende Werte**
In Deutschland geboren Ja: n=1128 Nein: n=57	1185	2
Sozioökonomischer Status Hoch (Fachabitur/Abitur/Studium): n=693 Niedrig (Niedrigere Ausbildung): n=479	1172	15
Atopie der Eltern Ja: n=536 Nein: n=639	1175	12
Anzahl Geschwister 0: n=189 1: n=654 2: n=216 3: n=66 4: n=18 5: n=8 6: n=1 7: n=2	1154	33
Als Säugling gestillt Ja: n=956 Nein: n=185	1141	46
Passivrauch Eltern Raucher: n=332 Eltern Ex-Raucher: n=95 Eltern Nichtraucher: n=737	1164	23
Studienzentrum Dresden: n=571 München: n=616	1187	0
Neurodermitis zum Zeitpunkt ISAAC II Ja: n=121 Nein: n=1066	1187	0
Allergische Rhinitis zum Zeitpunkt ISAAC II Ja: n=86 Nein: n=1101	1187	0
Asthma zum Zeitpunkt ISAAC II Ja: n=47 Nein: n=1140	1187	0

Tabelle 5.1: Fehlende Werte: Potentielle Confoundervariablen aus ISAAC II

Variablenbeschreibung und Häufigkeit	**vorhandene Werte**	**fehlende Werte**
Passivrauch Ja: n=723 Nein: n=455	1178	9
Rauchverhalten Raucher: n=341 Nichtraucher: n=839	1180	7
Geschlecht Männlich: n=480 Weiblich: n=707	1187	0
Neurodermitis zum Zeitpunkt SOLAR I Ja: n=137 Nein: n=1050	1187	0
Allergische Rhinitis zum Zeitpunkt SOLAR I Ja: n=158 Nein: n=1029	1187	0
Asthma zum Zeitpunkt SOLAR I Ja: n=43 Nein: n=1144	1187	0

Tabelle 5.2: Fehlende Werte: Potentielle Confoundervariablen aus SOLAR I

Variablenbeschreibung und Häufigkeit	**vorhandene Werte**	**fehlende Werte**
Passivrauch Ja: n=677 Nein: n=495	1172	15
Rauchverhalten Raucher: n=425 Ex-Raucher: n=99 Nichtraucher: n=657	1181	6
Schulbildung Höhere Schulbildung (Abi/FH): n=731 Niedrigere Schulbildung: n=452	1183	4
Allergische Rhinitis zum Zeitpunkt SOLAR II Ja: n=181 Nein: n=1006	1187	0
Asthma zum Zeitpunkt SOLAR II Ja: n=60 Nein: n=1127	1187	0

Tabelle 5.3: Fehlende Werte: Potentielle Confoundervariablen aus SOLAR II

Variablenbeschreibung und Häufigkeit	**vorhandene Werte**	**fehlende Werte**
Berufssituation - SOLAR I HauptschülerIn: n=8 RealschülerIn: n=137 GymnasiastIn: n=617 SchülerIn einer andern Schule: n=103 AuszubildendeR/BerufsschülerIn: n=297 StudentIn: n=3 Angestellt: n=6 Arbeitslos&-suchend: n=5 Sonstiges: n=10	1186	1
Berufssituation - SOLAR II AuszubildendeR/BerufsschülerIn: n=198 StudentIn (hauptberuflich): n=519 Angestellt: n=349 Selbstständig: n=12 Arbeitslos&-suchend: n=35 Aus gesundh. Gründen nicht arbeitend: n=2 Mutterschutz/Elternzeit/Beurlaubung: n=10 Sonstiges: n=60	1185	2

Tabelle 5.4: Fehlende Werte: Variablen als Zusatzinformation

5.1 Imputation durch Ziehen gemäß der Randverteilung der Daten

Bei dieser Methode wurden fehlende Werte ersetzt durch Werte, die gemäß der Randverteilung der beobachteten Daten gezogen wurden. Dabei wurde jede Variable separat betrachtet.
Um das Problem der Varianzunterschätzung zu vermeiden, das bei Single Imputation-Methoden, zu denen diese Methode zählt, häufig auftritt, wurde insgesamt zweimal aus der empirischen Randverteilung der Daten gezogen, so dass insgesamt zwei vervollständigte Datensätze resultierten. Zusätzlich wurden drei Datensätze mithilfe des R-Packages AMELIA II imputiert. So soll die Variabilität der Daten entsprechend berücksichtigt werden.

5.1.1 Binäre Variablen

Bei binären Variablen wurden die Werte, die für die fehlenden Werte eingesetzt wurden, gemäß einer Bernoulliverteilung gezogen. Hierbei wurde die Auftretenswahrscheinlichkeit einer 1, die zur Modellierung der Bernoulliverteilung notwendig ist, aus den beobachteten Daten ermittelt.

5.1.2 Kategoriale Variablen

Bei kategorialen Variablen wurden die Auftretenswahrscheinlichkeiten der jeweiligen Kategorien aus den beobachteten Daten für die entsprechende Variable ermittelt. Die Werte, durch welche die fehlenden Werte anschließend ersetzt wurden, wurden also aus einer Menge der für die jeweilige Variable möglichen Kategorien mit vorgegebenen Wahrscheinlichkeiten gezogen.

5.2 Imputation mithilfe des R-Packages AMELIA II

5.2.1 Allgemeines

Im Rahmen dieser Bachelorarbeit wurde für die Multiple Imputation das R-Package "AMELIA II: A Program for Missing Data" von J. Honaker, G. King und M. Blackwell verwendet. Diesem Package wird ein unvollständiger Datensatz übergeben, der imputiert werden soll. Die Imputation wird von dem Programmpaket durchgeführt und ausgegeben wird eine vom Benutzer festgelegte Anzahl (m) an vervollständigten Datensätzen. Der Vorteil dieses R-Packages gegenüber anderen Packages ist, dass mehr Variablen in die Imputation mit aufgenommen werden können und in kürzerer Zeit imputiert wird.

5.2.2 Wie funktioniert AMELIA II ?

Durch den in AMELIA II enthaltenen Algorithmus werden zuerst durch ein Bootstrap-Verfahren Datensätze erstellt, welche die gleiche Dimension wie der zu imputierende Datensatz aufweisen. Aus diesem Datensatz werden dann durch den EM-Algorithmus suffiziente Statistiken geschätzt und anschließend die fehlenden Werte des ursprünglichen Datensatzes imputiert. Dieses Verfahren wird nun m mal wiederholt, um m vervollständigte Datensätze zu erhalten, in denen die beobachteten Werte fest sind (exakt die gleichen wie im ursprünglichen Datensatz) und die fehlenden Werte durch den Algorithmus ersetzt werden.

Um Multiple Imputation anzuwenden, ist ein statistisches Modell erforderlich, aufgrund dessen m Imputationen für jeden fehlenden Wert im Datensatz berechnet werden. Ein Modell, dass sich für Probleme mit fehlenden Daten in einer überraschend großen Anzahl von Situationen als hilfreich erwiesen hat nimmt an, dass die Variablen gemeinsam multivariat normalverteilt sind [King et al. 2001].
Es wird also die Annahme getroffen, dass die vorliegenden Daten D multivariat normalverteilt sind mit Mittelwertsvektor μ und Varianz-Kovarianz-Matrix Σ: $D \sim N(\mu, \Sigma)$ [Honaker und King 2008]. Dann wird die Imputation der fehlenden Werte in folgenden Schritten durchgeführt:

- Durch Bootstrapping werden m Stichprobendatensätze der Größe n mit Zurücklegen aus dem Datensatz D gezogen. (Hierdurch entsteht Variabilität der Datensätze und somit wird die Unsicherheit bei der Imputation der fehlenden Werte berücksichtigt.)
- Auf jeden der m Stichprobendatensätze wird der EM-Algorithmus angewandt, um Punktschätzer für den Mittelwertsvektor μ und die Varianz-Kovarianz-Matrix Σ zu erhalten.

Funktionsweise des EM-Algorithmus

Der EM-Algorithmus ist ein iterativer Prozess, der aus den zwei aufeinanderfolgenden Schritten **E**xpectation-Schritt (E-Schritt) und **M**aximization-Schritt (M-Schritt) besteht. Im ersten E-Schritt wird für den Schätzer für $\theta = (\mu, \Sigma)$ ein Startwert angenommen. Aufgrund dieses Startwerts werden die Werte, durch welche die fehlenden Werte im Datensatz ersetzt werden, vorhergesagt. Im M-Schritt wird anschließend auf Basis der momentanen Version des so vervollständigten Datensatzes durch Maximum-Likelihood-Schätzung ein neuer Schätzer $\widehat{\theta}$ berechnet. Dieser neue Schätzer wird anschließend verwendet, um den E-Schritt erneut durchzuführen. Es werden also erneut

aufgrund des momentanen Schätzers $\widehat{\theta}$ die Werte, durch welche die fehlenden Werte ersetzt werden, vorhergesagt und der Datensatz somit vervollständigt. Anschließend wird durch den M-Schritt erneut ein Schätzer für θ berechnet und so weiter.
Jeder sequentielle Wert des Schätzers $\widehat{\theta}$ liegt somit notwendigerweise näher an der ML-Schätzer $\widehat{\theta}_{ML}$, der Schätzer nähert sich also immer weiter an $\widehat{\theta}_{ML}$ an. Wenn man nun genügend viele Iterationen vornimmt, kann man beliebig nah an den ML-Schätzer kommen. Üblicherweise wird der Prozess beendet, sobald Konvergenz vorliegt, d.h., wenn die Veränderung zwischen zwei aufeinanderfolgenden Werten von $\widehat{\theta}$ so klein erscheint, dass man annimmt, sich in einer ausreichend engen Nachbarschaft des Optimums zu befinden. Die Konvergenzrate ist dabei bestimmt von der Rate der fehlenden Werte im Datensatz. Gäbe es keine fehlenden Werte, würde der Algortihmus sofort konvergieren. Je mehr Werte im Datensatz fehlen, desto mehr Iterationen müssen durchgeführt werden.
Während aller Iterationen sind die Werte der beobachteten Daten konstant, nur die fehlenden Werte werden immer wieder neu ersetzt.

- Für jeden durch den EM-Algorithmus enstandenen Parameterschätzer $\widehat{\theta}_{ML}$ wird der Originaldatensatz (enthält noch die fehlenden Werte) verwendet, um die fehlenden Werte an ihren ursprünglichen Positionen zu ersetzen.

Hierdurch entstehen m multipel imputierte Datensätze, die für weitere Analysen verwendet werden können [Honaker und King 2008].

Die Annahme einer multivariaten Normalverteilung ist offensichtlich eine Approximation, da wenige Datensätze Variablen enthalten, die alle stetig und unbeschränkt sind, geschweige denn multivariat normalverteilt. Dennoch haben viele Wissenschaftler gezeigt, dass diese Approximation meist genauso gut funktioniert wie kompliziertere Alternativen, die speziell für kategoriale oder gemischte Daten entworfen wurden. Transformationen und andere Prozeduren können genutzt werden, um die Anpassung an das Modell zu verbessern [King et al. 2001]. Im R-Package AMELIA II werden zum Beispiel Transformationen von nominalen Variablen vorgenommen, um die Daten besser an ein solches Modell anzupassen.

5.2.3 Transformation von Variablen

In dem Programmpaket AMELIA II wird angenommen, dass die zu imputierenden Daten multivariat normalverteilt sind. AMELIA II kann aber auch direkt mit ordinalen oder nominalen Variablen umgehen. Es werden interne Transformationen dieser Variablen durchgeführt, um die Daten besser an die Annahmen der multivariaten Normalverteilung anzupassen. Am Ende werden vervollständigte Datensätze ausgegeben, welche die Variablen in ihrer ursprünglichen Form (ursprüngliches Skalenniveau) enthalten. Durch

eine Rücktransformation erhält man also die Datensätze so zurück, wie man sie dem Programm zu Beginn übergeben hat.

Transformation nominaler Variablen

Nominale Variablen, die bei der Imputation explizit als solche behandelt werden sollen, müssen in AMELIA II angegeben werden. Für eine multinomial verteilte Variable mit p Kategorien wird zuerst p (die Anzahl der Kategorien) bestimmt. Anschließend werden p-1 binäre Variablen erstellt, um jede mögliche Kategorie zu spezifizieren. Für diese binären Variablen werden dann stetige Werte für die Imputation erstellt. Diese Werte werden geeignet in Wahrscheinlichkeiten für jede der p möglichen Kategorien transformiert. Eine der Kategorien wird dann gezogen, wodurch die p-kategoriale, multinomial verteilte Variable wieder hergestellt wird, die dann ausgegeben wird.

In dieser Arbeit wurden sämtliche potentielle Confoundervariablen dem Package AMELIA II als nominale Variablen übergeben. Diese Variablen waren: Studienzentrum, Geschlecht, Sozioökonomischer Status, In Deutschland geboren, Atopie der Eltern, Anzahl der Geschwister, Gestillt, Passivrauch (aus ISAAC II, SOLAR I und SOLAR II), Schulbildung, Rauchverhalten (aus SOLAR I und SOLAR II), Neurodermitis (aus ISAAC II, SOLAR), Allergische Rhinitis (aus ISAAC II, SOLAR I und SOLAR II) und Asthma (aus ISAAC II, SOLAR I und SOLAR II).
Als Zusatzinformation für die Imputation wurden die Variablen Berufssituation zum Zeitpunkt SOLAR I und Berufssituation zum Zeitpunkt SOLAR II verwendet. Diese Variablen wurden nur als zusätzliche Informationsquelle für die Imputation verwendet, gehen aber nicht in die später gerechneten logistischen Modelle in dieser Arbeit ein.

5.2.4 Identifikationsvariablen

Als Identifikationsvariablen können im Package AMELIA II Variablen angegeben werden, die nicht für die Imputation verwendet werden, aber trotzdem im Datensatz verbleiben sollen. Bei der Imputation der fehlenden Werte in den potentiellen Confoundervariablen wurde die Kohortennummer der Probanden (knr), durch die jeder Proband eindeutig identifiziert werden kann, als Identifikationsvariable aufgenommen.

5.2.5 Auswahl der Variablen bei der Imputation

Bei der Imputation spielt die Auswahl der Variablen, die für die Imputation verwendet werden, eine wichtige Rolle. Generell geht man dabei so vor, dass für die Imputation zumindest die Variablen verwendet werden, die auch später in die Datenanalyse (z.B. in

Regressionsmodelle) eingehen sollen [King et al. 2001]. Lässt man bei der Imputation eine Variable weg, die bei der Analyse des Datensatzes berücksichtigt wird, so kann dies schwerwiegende Folgen haben. Schätzer, welche die Beziehung zwischen dieser (weggelassenen) Variable und anderen Variablen messen, werden gegen null verzerrt.
Gemäß dieser Richtlinie wurden in dieser Arbeit alle Variablen, die als mögliche Kovariablen für Regressionsmodelle im Rahmen der Datenanalyse in Betracht gezogen wurden und auch die vollständigen Zielvariablen ("Allergische Rhinitis in SOLAR II" und "Asthma in SOLAR II") für die Imputation der fehlenden Werte verwendet.

5.2.6 Behandlung von Variablen mit hohen Korrelationen

Bestehen im Datensatz hohe Korrelationen zwischen den Variablen, so wird empfohlen, einen sogenannten Ridge Prior hinzuzufügen. Durch diesen Ridge Prior wird numerische Stabilität erreicht, in dem man die Kovarianzen zwischen den Variablen gegen Null schrumpfen lässt, ohne die Mittelwerte oder Varianzen zu verändern. Eine positive Zahl (x) als Prior anzugeben entspricht grob gesehen dem Hinzufügen von x künstlichen Beobachtungen zum Datensatz, welche die gleichen Mittelwerte und die gleichen Varianzen besitzen wie die echten beobachteten Daten, aber mit Kovarianzen gleich Null. Es wird empfohlen, die Anzahl der künstlich hinzugefügten Beobachtungen möglichst klein zu halten, und sie nur zu erhöhen, wenn es wirklich notwendig ist. Der Startwert für die Anzahl der hinzugefügten Beobachtungen sollte bei 0,5 bis 1 % der Anzahl der im Datensatz enthaltenen echten Beobachtungen liegen.
Bei der Imputation der fehlenden Werte in den potentiellen Confoundervariablen in dieser Arbeit wurden für den Datensatz mit 1.187 Beobachtungen x=5,9 ($\approx$ 0,5 %) künstliche Beobachtungen gewählt [Honaker et al. 2009].

5.3 Übersicht über die Variablenausprägungen in den imputierten Datensätzen

Durch die beiden Methoden Imputation durch Ziehen gemäß der Randverteilung der Daten und Multiple Imputation mithilfe des R-Pakets AMELIA II wurden insgesamt fünf vervollständigte Datensätze für den unvollständigen Datensatz, der die Confoundervariablen enthält, erstellt. Abbildung 5.2 gibt eine Übersicht über die Ausprägungen der Variablen in den imputierten Datensätzen.

Variablen-beschreibung	Ausprägung	ursprüngl. Datensatz	Amelia 1. Datensatz	Amelia 2. Datensatz	Amelia 3. Datensatz	empir. Vert. 1. Datensatz	empir. Vert. 2. Datensatz
In Deutschland geboren	Nein(0)	57	57	57	57	57	57
	Ja (1)	1128	1130	1130	1130	1130	1130
	Fehlende Angabe(NA)	2	0	0	0	0	0
Atopie der Eltern	Nein(0)	639	646	650	646	646	642
	Ja (1)	536	541	537	541	541	545
	Fehlende Angabe(NA)	12	0	0	0	0	0
Geschwister	Nein(0)	189	193	193	193	194	196
	Ja (1)	965	994	994	994	993	991
	Fehlende Angabe(NA)	33	0	0	0	0	0
Kind gestillt	Nein(0)	185	199	196	197	193	193
	Ja (1)	956	988	991	990	994	994
	Fehlende Angabe(NA)	46	0	0	0	0	0
Passivrauch (ISAAC II)	Eltern Nichtraucher (0)	737	751	745	752	751	749
	Eltern Raucher (1)	332	339	344	337	338	341
	Eltern Ex-Raucher (2)	95	97	98	98	98	97
	NA	23	0	0	0	0	0
Passivrauch (SOLAR I)	Nein(0)	455	459	458	459	459	461
	Ja (1)	723	728	729	728	728	726
	Fehlende Angabe(NA)	9	0	0	0	0	0
Rauchverhalten (SOLAR I)	Nichtraucher (0)	839	843	845	844	843	841
	Raucher (1)	341	344	342	343	344	346
	Fehlende Angabe(NA)	7	0	0	0	0	0
Passivrauch (SOLAR II)	Nein(0)	495	497	496	499	502	504
	Ja (1)	677	690	691	688	685	683
	Fehlende Angabe(NA)	15	0	0	0	0	0
Rauchverhalten (SOLAR II)	Nichtraucher(0)	657	659	659	658	661	658
	Raucher (1)	425	427	427	429	427	429
	Ex-Raucher (2)	99	101	101	100	99	100
	Fehlende Angabe(NA)	6	0	0	0	0	0
Schulbildung (SOLAR II)	Niedrigere (0)	452	454	456	456	454	455
	Höhere (Abi/FH) (1)	731	733	731	731	733	732
	Fehlende Angabe(NA)	4	0	0	0	0	0
Sozioökonomischer Status	Niedrig (0)	479	486	490	486	485	488
	Hoch (1)	693	701	697	701	702	699
	Fehlende Angabe(NA)	15	0	0	0	0	0
Berufssituation (SOLAR I)	Hauptschüler (1)	8	8	8	8	8	8
	Realschüler (2)	137	138	137	137	138	137
	Gymnasiast (3)	617	617	617	617	617	617
	Schüler andere Schule (4)	103	103	103	104	103	104
	Azubi/Berufsschüler (5)	297	297	298	297	297	297
	Student (6)	3	3	3	3	3	3
	Angestellt (7)	6	6	6	6	6	6
	Arbeitslos&-suchend(9)	5	5	5	5	5	5
	Sonstiges (12)	10	10	10	10	10	10
	Fehlende Angabe(NA)	1	0	0	0	0	0
Berufssituation (SOLAR II)	Azubi/Berufsschüler (1)	198	198	198	198	199	199
	Student (2)	519	520	520	520	520	519
	Angestellt (3)	349	349	350	350	349	349
	Selbstständig (4)	12	12	12	12	12	12
	Arbeitslos&-suchend (5)	35	35	35	35	35	35
	aus gesundh. Gründen nicht arbeitend (6)	2	2	2	2	2	2
	Mutterschutz/Elternzeit/Beurlaubung (8)	10	11	10	10	10	10
	Sonstiges (9)	60	60	60	60	60	61
	Fehlende Angabe(NA)	2	0	0	0	0	0

Abbildung 5.2: Variablenausprägungen in den imputierten Datensätzen

KAPITEL 6

Berechnung der Expositionsvariablen

In den Fragebögen für die Studien SOLAR I und SOLAR II wurden von den Probanden Angaben zu ausgeübten Tätigkeiten gemacht. Dabei konnte jeder Proband pro Studie bis zu fünf Tätigkeitsangaben machen. Diese Tätigkeitsangaben umfassten Angaben zur ausgeübten Tätigkeit, zur Branche in der gearbeitet wurde, zu Beginn und Ende der Tätigkeit (Anfangsmonat, Anfangsjahr, Endmonat und Endjahr) und zur Anzahl der Wochenstunden, die gearbeitet wurden.
Durch die Kodierung der Tätigkeiten anhand des ISCO 88-Codes konnte jeder angegebenen Tätigkeit ein vierstelliger Code zugeordnet werden. Mithilfe dieses Codes konnte dann für jede Tätigkeit festgelegt werden, ob eine Exposition vorlag und wenn ja in welcher Kategorie der Job-Exposure-Matrix diese vorlag. Für die Berechnung der Expositionen mithilfe der Job-Exposure-Matrix (JEM) wurden die Unterkategorien der JEM zu fünf Oberkategorien (HMW, LMW, MIXED,IRRPEAKS, LOWRISK) zusammengefasst. Dabei lag in einer Oberkategorie eine Exposition vor, wenn in mindestens einer ihrer Unterkategorien eine Exposition vorlag. Die Zuordnung der Unterkategorien zu den fünf Oberkategorien ist Abbildung 2.3 in Kapitel 2 zu entnehmen.

6.1 Komplexe Matrix als Basis für alle Expositionsberechnungen

Die ursprüngliche Anforderung bestand darin, auf Basis der Probanden, die vollständige Tätigkeitsangaben gemacht haben, zuerst eine jahreweise aufgetrennte Matrix zu erstellen, aus der die kumulierte Exposition über alle Tätigkeiten und Jahre, die Exposition im ersten Tätigkeitsjahr (12 Monate ab Beginn der ersten Tätigkeit) sowie die Exposition in der ersten ausgeübten Tätigkeit berechnet werden konnten.
Von den 1093 Probanden, die vollständige Tätigkeitsangaben gemacht haben, hatten 25 Probanden bereits vor dem Jahr 2000 gearbeitet. Da dies nur ein kleiner Anteil ist, wurde die Matrix nur für diese Probanden für die Jahre 1992 bis 2009 erstellt. Für die

restlichen 1068 Probanden, die noch nicht vor dem Jahr 2000 gearbeitet hatten, wurde die Matrix für die Jahre 2000 bis 2009 erstellt.
Um die Erstellung der Matrix, die als Basis für alle Expositionsberechnungen dienen sollte sowie die weiteren Schritte, die zur Berechnung der Expositionen nötig waren, beispielhaft darzustellen, wird hier jeweils in kursiver Schrift zu jedem Schritt ein Beispiel ausgeführt.

Als Grundlage für die Erstellung der Matrix dienten jeweils die Tätigkeitsangaben, die der entsprechende Proband gemacht hatte.

Ein Proband hat Angaben zu drei Tätigkeiten (Pflegepraktikum, Freiwilliges soziales Jahr und Ausbildung zur Gesundheits- und Krankenpflegerin) gemacht. Die ISCO-Codes 8888 und 9999 in den restlichen sieben Zeilen stehen dafür, dass keine Tätigkeitsangabe gemacht wurde. Im Pflegepraktikum wurde von Juli 2004 bis August 2004 gearbeitet. Im Freiwilligen sozialen Jahr wurde von September 2004 bis Februar 2005 gearbeitet und in der Ausbildung zur Gesundheits- und Krankenpflegerin von März 2005 bis Februar 2008. Es wurden jeweils 40 Stunden pro Woche gearbeitet. Alle drei Tätigkeiten wurden mit dem ISCO-Code 5132 kodiert. Es bestand somit in allen drei Tätigkeiten Exposition in den Kategorien HMW und LMW.

knr	Anf.monat	Anf.jahr	Endmonat	Endjahr	Wstd.	Isco	HMW	LMW	MIXED	IRRPEAKS	LOWRISK
D56657290	7	2004	8	2004	40	5132	1	1	0	0	0
D56657290	9	2004	2	2005	40	5132	1	1	0	0	0
D56657290	3	2005	2	2008	40	5132	1	1	0	0	0
D56657290	NA	NA	NA	NA	NA	8888	0	0	0	0	0
D56657290	NA	NA	NA	NA	NA	8888	0	0	0	0	0
D56657290	NA	NA	NA	NA	NA	9999	0	0	0	0	0
D56657290	NA	NA	NA	NA	NA	9999	0	0	0	0	0
D56657290	NA	NA	NA	NA	NA	9999	0	0	0	0	0
D56657290	NA	NA	NA	NA	NA	9999	0	0	0	0	0
D56657290	NA	NA	NA	NA	NA	9999	0	0	0	0	0

Abbildung 6.1: Beispiel: Tätigkeitsangaben eines Probanden

Die Tätigkeitsangaben konnten nun in einer komplexen Matrix dargestellt werden. Folgende Variablen sind in dieser Matrix enthalten:

- die Kohortennummer des Probanden (knr)
- die Nummer der Tätigkeit (Nr_Beruf); die Tätigkeiten wurden hier chronologisch geordnet und die Tätigkeit mit der Nummer 1 ist dabei die erste ausgeübte Tätigkeit des Probanden
- für jede Tätigkeit die Jahre 2000-2009 (bzw. bei den Probanden, die schon vor 2000 gearbeitet hatten, die Jahre 1992-2009)
- der Anfangsmonat (Anf.monat) der Tätigkeit im entsprechenden Jahr
- der Endmonat der Tätigkeit im entsprechenden Jahr
- Anzahl der Wochenstunden (Wstd.), die in der entsprechenden Tätigkeit gearbeitet wurden
- ISCO-Code der Tätigkeit
- HMW: Gibt an, ob Exposition in der Kategorie HMW bestand (0=nein, 1=ja) (Analog für LMW,MIXED,IRRPEAKS und LOWRISK)
- Gearb. Monate: Gibt an, wie viele Monate im entsprechenden Jahr (Zeile) gearbeitet wurden
- HMW_jahr: Exposition in der Kategorie HMW pro Jahr:
  Wochenstunden $\cdot$ 4,25 $\cdot$ HMW $\cdot$ gearbeitete Monate
  (Analog für LMW,MIXED,IRRPEAKS und LOWRISK)

Die Anzahl der Wochenstunden, alle Expositionen und die im entsprechenden Jahr gearbeiteten Monate wurden nur in den Jahren (Zeilen) eingetragen, in denen auch wirklich gearbeitet wurde.

Für die Berechnung der Expositionen pro Monat wurden zuerst die Wochenstunden mit 4,25 multipliziert, um die gearbeiteten Stunden pro Monat zu erhalten. Anschließend wurden die Einträge der JEM (0 oder 1) mit den gearbeiteten Stunden pro Monat multipliziert, um die Exposition pro Monat zu erhalten. Um die Expositionen pro Jahr zu berechnen, wurden die Expositionen pro Monat mit den gearbeiteten Monaten im entsprechenden Jahr multipliziert.

Berechnung der Exposition pro Jahr (z.B. HMW_jahr):
HMW_jahr = 4.25 · Wochenstunden · HMW · gearbeitete Monate

Dabei wurden diese Berechnungen nur durchgeführt, wenn in der jeweiligen Tätigkeit mindestens acht Wochenstunden gearbeitet wurde. Wurde weniger als acht Stunden pro Woche gearbeitet, so wurden die jährlichen Expositionen auf 0 (nicht exponiert) gesetzt. Der Anfangsmonat der Tätigkeit wurde in die Zeile eingetragen, die das Anfangsjahr enthält. Der Endmonat der Tätigkeit wurde in die Zeile eingetragen, die das Endjahr enthält. Waren Anfangsmonat und Endmonat im gleichen Jahr, so wurden sie in die gleiche Zeile eingetragen. War der Endmonat in einem späteren Jahr als der Anfangsmonat, so wurden die Anfangsmonate zwischen Anfangs- und Endjahr mit einer 1 (für Januar) und die Endmonate mit einer 12 (für Dezember) ausgefüllt.
Alle anderen Monatsangaben für die entsprechende Tätigkeit wurden mit einer 0 ausgefüllt. Das war dann der Fall, wenn das Jahr in der jeweiligen Zeile später war als das Endjahr der Tätigkeit, da dann in diesem Jahr nicht mehr gearbeitet wurde, oder wenn das Jahr in der jeweiligen Zeile früher war als das Anfangsjahr der Tätigkeit, da dann in diesem Jahr noch nicht gearbeitet wurde.
Probanden mit ISCO-Code 94,95,98 bei einer Tätigkeit konnte bei der Tätigkeitskodierung durch den ISCO-Code keine eindeutige Exposition zugewiesen werden und alle Expositionen in der JEM wurden deshalb auf 0 (nicht exponiert) gesetzt. Probanden mit ISCO-Code 97 hatten zum Zeitpunkt der enstprechenden Studie noch keine Exposition und alle Expositionen in der JEM wurden deswegen auf 0 (nicht exponiert) gesetzt. Probanden mit ISCO-Code 9999 bzw. 8888 bei einer Tätigkeit hatten nicht gearbeitet und hatten demzufolge auch keine Exposition. Es wurden für die Expositionsberechnungen außerdem nur Probanden betrachtet, die in der entsprechenden Tätigkeit mehr als acht Wochenstunden gearbeitet hatten. Bei den Probanden mit weniger als acht Wochenstunden musste also später auch nicht berechnet werden, wie viele Monate sie gearbeitet hatten, da alle monatlichen und jährlichen Expositionen auf 0 (nicht exponiert) gesetzt wurden.

Da das Pflegepraktikum (Nr_Beruf = 1) im Juli 2004 begann und im August 2004 endete, wurden Anfangsmonat und Endmonat in dieselbe Zeile eingetragen (Jahr 2004). Da das Freiwillige soziale Jahr (Nr_Beruf = 2) im September 2004 begann und im Februar 2005 endete, wurde der Anfangsmonat in die Zeile mit dem Jahr 2004 eingetragen, der Endmonat im Jahr 2004 ist Dezember, da ja bis 2005 gearbeitet wurde. Im Jahr 2005 wurde als Endmonat Februar eingetragen und als Anfangsmonat Januar, da bis Februar gearbeitet wurde, also der Anfangsmonat im Jahr 2005 der Januar war. Bei der Ausbildung zur Gesundheits- und Krankenpflegerin (Nr_Beruf = 3) wurde analog vorgegangen.

Zudem wurde hier in den Jahren zwischen dem Anfangsjahr und dem Endjahr jeweils beim Anfangsmonat der Januar und beim Endmonat der Dezember eingetragen, da in diesen Jahren voll gearbeitet wurde. Pro Jahr wurde anschließend berechnet, wie viele Monate gearbeitet wurden. Im Pflegepraktikum wurden im Jahr 2004 beispielsweise 2 Monate gearbeitet. Die Expositionen pro Monat und pro Jahr wurden nur in denjenigen Zeilen berechnet, in denen auch wirklich gearbeitet wurde. Berechnung der Expositionen (beispielhafte Berechnung nur für die erste Tätigkeit des Probanden):
$HMW\_jahr = 4{,}25 \cdot 40\ Wochenstunden \cdot HMW \cdot 2\ Monate = 340,$
$LMW\_jahr = 4{,}25 \cdot 40\ Wochenstunden \cdot LMW \cdot 2\ Monate = 340.$

knr	Nr_Beruf	Jahr	Anf.monat	Endmonat	Wstd.	ISCO	Gearb.Monate	HMW_jahr	LMW_jahr	MIXED_jahr	IRRPEAKS_jahr	LOWRISK_jahr
D56657290	1	2000	0	0	0	5132	0	0	0	0	0	0
D56657290	1	2001	0	0	0	5132	0	0	0	0	0	0
D56657290	1	2002	0	0	0	5132	0	0	0	0	0	0
D56657290	1	2003	0	0	0	5132	0	0	0	0	0	0
D56657290	1	2004	7	8	40	5132	2	340	340	0	0	0
D56657290	1	2005	0	0	0	5132	0	0	0	0	0	0
D56657290	1	2006	0	0	0	5132	0	0	0	0	0	0
D56657290	1	2007	0	0	0	5132	0	0	0	0	0	0
D56657290	1	2008	0	0	0	5132	0	0	0	0	0	0
D56657290	1	2009	0	0	0	5132	0	0	0	0	0	0
D56657290	2	2000	0	0	0	5132	0	0	0	0	0	0
D56657290	2	2001	0	0	0	5132	0	0	0	0	0	0
D56657290	2	2002	0	0	0	5132	0	0	0	0	0	0
D56657290	2	2003	0	0	0	5132	0	0	0	0	0	0
D56657290	2	2004	9	12	40	5132	4	680	680	0	0	0
D56657290	2	2005	1	2	40	5132	2	340	340	0	0	0
D56657290	2	2006	0	0	0	5132	0	0	0	0	0	0
D56657290	2	2007	0	0	0	5132	0	0	0	0	0	0
D56657290	2	2008	0	0	0	5132	0	0	0	0	0	0
D56657290	2	2009	0	0	0	5132	0	0	0	0	0	0
D56657290	3	2000	0	0	0	5132	0	0	0	0	0	0
D56657290	3	2001	0	0	0	5132	0	0	0	0	0	0
D56657290	3	2002	0	0	0	5132	0	0	0	0	0	0
D56657290	3	2003	0	0	0	5132	0	0	0	0	0	0
D56657290	3	2004	0	0	0	5132	0	0	0	0	0	0
D56657290	3	2005	3	12	40	5132	10	1700	1700	0	0	0
D56657290	3	2006	1	12	40	5132	12	2040	2040	0	0	0
D56657290	3	2007	1	12	40	5132	12	2040	2040	0	0	0
D56657290	3	2008	1	2	40	5132	2	340	340	0	0	0
D56657290	3	2009	0	0	0	5132	0	0	0	0	0	0

Abbildung 6.2: Beispiel: Komplexe Matrix zur Expositionsberechnung

6.2 Berechnung der Exposition kumuliert über alle Tätigkeiten und Jahre

Um die Exposition kumuliert über alle Tätigkeiten und Jahre zu erhalten, wurde pro Proband die Exposition in jeder der fünf Kategorien der JEM (HMW_jahr, LMW_jahr, MIXED_jahr, IRRPEAKS_jahr, LOWRISK_jahr) über alle Tätigkeiten und Jahre aufsummiert. Es wurde zusätzlich auch eine binäre Variable erstellt, die angibt, ob der Proband in der jeweiligen Kategorie der JEM mindestens einmal exponiert war oder nicht.
Die so erhaltenen Variablen können direkt als Einflussgrößen in die logistische Regression eingehen.

HMW_kumuliert = 340 + 680 + 340 + 1700 + 2040 + 2040 + 340 = 7480,
LMW_kumuliert = 340 + 680 + 340 + 1700 + 2040 + 2040 + 340 = 7480.
HMW_binaer = 1, LMW_binaer = 1.

knr	HMW_kumuliert	LMW_kumuliert	MIXED_kumuliert	IRRPEAKS_kumuliert	LOWRISK_kumuliert
D56657290	7480	7480	0	0	0
knr	**HMW_binaer**	**LMW_binaer**	**MIXED_binaer**	**IRRPEAKS_binaer**	**LOWRISK_binaer**
D56657290	1	1	0	0	0

Abbildung 6.3: Beispiel: Exposition kumuliert über alle Tätigkeiten und Jahre

6.3 Berechnung der Exposition in der ersten ausgeübten Tätigkeit

Die Exposition in der ersten ausgeübten Tätigkeit wurde betrachtet, da in Bezug auf Atemwegserkrankungen gezeigt werden konnte, dass sich die Exposition in der ersten ausgeübten Tätigkeit als Surrogat für die Exposition aller ausgeübter Tätigkeiten eignen kann [Benke et al. 2008].

Um die Exposition in der ersten ausgeübten Tätigkeit zu erhalten, wurde für jeden Probanden nur die erste Tätigkeit (Nr_Beruf=1) aus der komplexen Matrix betrachtet. Dann wurden pro Proband nur für seine erste Tätigkeit die Expositionen über alle Jahre aufsummiert, um die gesamte Exposition in der ersten Tätigkeit zu erhalten. Zusätzlich wurde pro Kategorie der JEM eine binäre Variable erstellt, die angibt, ob der Proband in der ersten Tätigkeit in der jeweiligen Kategorie exponiert war oder nicht.
Die so erhaltenen Variablen können direkt als Einflussgrößen in die logistische Regression eingehen.

Im Beispiel: HMW_ersterberuf_gesamt = 340 und LMW_ersterberuf_gesamt = 340. HMW_ersterberuf_binaer = 1 und LMW_ersterberuf_binaer = 1.

knr	HMW_ersterberuf_gesamt	LMW_ersterberuf_gesamt	MIXED_ersterberuf_gesamt	IRRPEAKS_ersterberuf_gesamt	LOWRISK_ersterberuf_gesamt
D56657290	340	340	0	0	0
knr	**HMW_ersterberuf_binaer**	**LMW_ersterberuf_binaer**	**MIXED_ersterberuf_binaer**	**IRRPEAKS_ersterberuf_binaer**	**LOWRISK_ersterberuf_binaer**
D56657290	1	1	0	0	0

Abbildung 6.4: Beispiel: Exposition in der ersten ausgeübten Tätigkeit

6.4 Berechnung der Exposition im ersten Tätigkeitsjahr

Die Exposition im ersten Tätigkeitsjahr wurde betrachtet, da gezeigt werden konnte, dass Expositionen am Anfang des Berufslebens die Entstehung von berufsbedingtem Asthma tendenziell stärker beeinflussen, als spätere Expositionen [Benke et al. 2008].

Um die Exposition im ersten Tätigkeitsjahr zu berechnen, wurde zunächst ausgehend vom Beginn der ersten Tätigkeit des jeweiligen Probanden der 12-Monats-Zeitraum ermittelt, der dem ersten Tätigkeitsjahr entspricht. Das erste Tätigkeitsjahr beginnt also mit dem Beginn der ersten Tätigkeit und endet genau 12 Monate später.
Bei jeder Tätigkeit eines Probanden wurde nun überprüft, ob sie sich im 12-Monatszeitraum des ersten Tätigkeitsjahrs befindet oder nicht. Anschließend wurde bei den Tätigkeiten, die sich in diesem Zeitraum befanden, berechnet, wieviele Monate noch in diesen 12-Monatszeitraum fallen.
Durch das Aufsummieren der gearbeiteten Monate pro Tätigkeit über alle Jahre wurde die Anzahl der gearbeiteten Monate in der jeweiligen Tätigkeit berechnet. Anschließend wurde abgeglichen, wie viele Monate in der Tätigkeit insgesamt gearbeitet wurden und wie viele Monate noch in den 12-Monatszeitraum des ersten Tätigkeitsjahrs fallen. Die Anzahl der Monate, die für das erste Tätigkeitsjahr noch beachtet werden müssen, ist dann jeweils das Minimum der Monate, die noch in den 12-Monatszeitraum fallen und der Monate, die insgesamt in der Tätigkeit gearbeitet wurden.

Anschließend wurden nur bei den Tätigkeiten, die noch in den 12-Monatszeitraum des ersten Tätigkeitsjahrs fallen, die Expositionen berechnet. Dabei wurden für die Berechnung der Expositionen pro Monat zuerst die Wochenstunden mit 4,25 multipliziert, um die gearbeiteten Stunden pro Monat zu erhalten. Dann wurden die Einträge der JEM (0 oder 1) mit den gearbeiteten Stunden pro Monat multipliziert, um die Exposition pro Monat zu erhalten. Um die Expositionen pro Jahr zu berechnen, wurden die Expositionen pro Monat mit der Anzahl der Monate, die im entsprechenden Jahr für das erste

Tätigkeitsjahr beachtet werden müssen, multipliziert.
Um die gesamte Exposition im ersten Tätigkeitsjahr zu erhalten, wurde pro Proband für jedes Jahr die zu beachtende Exposition in jeder der fünf Kategorien der JEM über alle Tätigkeiten aufsummiert.
Es wurde zusätzlich eine binäre Variable erstellt, die angibt, ob der Proband im ersten Tätigkeitsjahr in der jeweiligen Kategorie der JEM mindestens einmal exponiert war oder nicht.
Die so erhaltenen Variablen können direkt als Einflussgrößen in die logistische Regression eingehen.

Im Beispiel: Der 12-Monats-Zeitraum für das erste Tätigkeitsjahr beginnt im Juli 2004 (Beginn der ersten Tätigkeit) und endet im Juni 2005. Das Pflegepraktikum (Nr_Beruf = 1) musste somit für das erste Tätigkeitsjahr komplett berücksichtigt werden. Da im Pflegepraktikum 2 Monate lang gearbeitet wurde ging die Exposition für die Kategorien HMW und LMW folgendermaßen ein:
4,25 · 40 Wochenstunden · HMW · 2 Monate = 340 bzw.
4,25 · 40 Wochenstunden · LMW · 2 Monate = 340.
Da das Freiwillige soziale Jahr (Nr_Beruf = 2) im September 2004 begann und im Februar 2005 endete, musste es somit auch komplett für das erste Tätigkeitsjahr berücksichtigt werden. Die Expositionen für die Kategorien HMW und LMW wurden folgendermaßen berechnet:
4,25 · 40 Wochenstunden · HMW · 4 Monate + 4,25 · 40 Wochenstunden · HMW · 2 Monate = 1020 bzw.
4,25 · 40 Wochenstunden · LMW · 4 Monate + 4,25 · 40 Wochenstunden · HMW · 2 Monate = 1020.
Da die Ausbildung zur Gesundheits- und Krankenpflegerin (Nr_Beruf = 3) im März 2005 begann und im Februar 2008 endete und das Ende des ersten Tätigkeitsjahrs Juni 2005 war, gingen von dieser Tätigkeit nur noch die vier Monate März bis Juni 2005 in das erste Tätigkeitsjahr ein. Die Expositionen für die Kategorien HMW und LMW wurden also folgendermaßen berechnet:
4,25 · 40 Wochenstunden · HMW · 4 Monate = 680 bzw.
4,25 · 40 Wochenstunden · LMW · 4 Monate = 680.
Die gesamte Exposition für das erste Tätigkeitsjahr in den Kategorien HMW und LMW wurde wie folgt berechnet:
HMW_erstesjahr_gesamt = 340 + 1020 + 680 = 1020,
LMW_erstesjahr_gesamt = 340 + 1020 + 680 = 2040.
HMW_erstesjahr_binaer = 1 und LMW_erstesjahr_binaer = 1.

knr	HMW_erstesjahr_gesamt	LMW_erstesjahr_gesamt	MIXED_erstesjahr_gesamt	IRRPEAKS_erstesjahr_gesamt	LOWRISK_erstesjahr_gesamt
D56657290	2040	2040	0	0	0
knr	**HMW_erstesjahr_binaer**	**LMW_erstesjahr_binaer**	**MIXED_erstesjahr_binaer**	**IRRPEAKS_erstesjahr_binaer**	**LOWRISK_erstesjahr_binaer**
D56657290	1	1	0	0	0

Abbildung 6.5: Beispiel: Exposition im ersten Tätigkeitsjahr

6.5 Betrachtung der gebildeten Expositionsvariablen

Übersicht über die erstellten Expositionsvariablen

Die folgende Tabelle gibt eine Übersicht über die in diesem Kapitel erstellten Expositionsvariablen.

Erstellte Expositionsvariablen:
- ◇ Kumulierte Exposition über alle Tätigkeiten und Jahre
- ◇ Binäre Exposition über alle Tätigkeiten und Jahre
- ◇ Kumulierte Exposition in der ersten ausgeübten Tätigkeit
- ◇ Binäre Exposition in der ersten ausgeübten Tätigkeit
- ◇ Kumulierte Exposition im ersten Tätigkeitsjahr
- ◇ Binäre Exposition im ersten Tätigkeitsjahr

Bemerkung

Zur Berechnung der Expositionsvariablen wurde in dieser Bachelorarbeit zusätzlich eine einfachere äquivalente Version der komplexen Matrix zur Expositionsberechnung erstellt, wobei hier eine jahreweise Auftrennung nicht nötig war. Die Expositionsvariablen konnten direkt auf Basis des Datensatzes, der die Tätigkeitsangaben der Probanden mit vollständigen Tätigkeitsangaben in zeilenweiser Form (pro Proband zehn Zeilen für maximal zehn Tätigkeiten in SOLAR I und SOLAR II) enthält, berechnet werden. Ein Vorteil gegenüber der zuerst erstellten komplexen Matrix war auch die sehr viel geringere Dauer, die zur Erstellung der Expositionsvariablen nötig war.
Will man jedoch beispielsweise die Exposition bis zu einem bestimmten Beobachtungszeitpunkt betrachten, d.h., zum Beispiel die Exposition bis zum Jahr 2005, so wird die Berechnung anhand der komplexen Matrix sehr viel einfacher sein als anhand der einfacheren äquivalenten Version, was ein Vorteil dieser komplexen Matrix ist.

Betrachtung der kumulierten Expositionsvariablen anhand von Tabellen und Boxplots

Die in den Abschnitten 6.2, 6.3 und 6.4 gebildeten Expositionsvariablen werden in den folgenden drei Abschnitten anhand von Tabellen und Boxplots dargestellt. Die Boxplots wurden jeweils nur auf Basis der Daten derjenigen Probanden erstellt, die in der jeweiligen Kategorie exponiert waren.

Exposition über alle Tätigkeiten und Jahre

Der Tabelle 6.1 ist das Auftreten der Expositionen über alle Tätigkeiten und Jahre hinweg sowie Median, Minimum und Maximum der berechneten Expositionen zu entnehmen. Am häufigsten trat eine Exposition in der Kategorie LOWRISK auf, gefolgt von den Kategorien LMW und HMW. Expositionen in der Kategorie MIXED und vor allem in der Kategorie IRRPEAKS traten sehr selten auf.
Vergleicht man die Mediane der Expositionen in den verschiedenen Kategorien, so lag die längste Exposition in den Kategorien IRRPEAKS und LMW vor, gefolgt von der Kategorie HMW. Die Mediane der Expositionen in den Kategorien MIXED und LOWRISK lagen deutlich darunter.
Von den Probanden, die keiner Exposition ausgesetzt waren, hatten 426 nie gearbeitet.

kumulierte Exposition	**Exposition vorhanden**	**Keine Exposition vorhanden**
HMW	Anzahl Fälle: 165 Median: 1360 Stunden Range in Stunden: [68,12240]	Anzahl Fälle ohne Exposition: 929 nie gearbeitet: 426 gearbeitet, aber keine Exposition: 503
LMW	Anzahl Fälle: 177 Median: 1530 Stunden Range in Stunden: [34,13090]	Anzahl Fälle ohne Exposition: 917 nie gearbeitet: 426 gearbeitet, aber keine Exposition: 491
MIXED	Anzahl Fälle: 34 Median: 669 Stunden Range in Stunden: [68,12240]	Anzahl Fälle ohne Exposition: 1060 nie gearbeitet: 426 gearbeitet, aber keine Exposition: 634
IRRPEAKS	Anzahl Fälle: 7 Median: 1530 Stunden Range in Stunden: [170,7990]	Anzahl Fälle ohne Exposition: 1087 nie gearbeitet: 426 gearbeitet, aber keine Exposition: 661
LOWRISK	Anzahl Fälle: 241 Median: 850 Stunden Range in Stunden: [34,15980]	Anzahl Fälle ohne Exposition: 853 nie gearbeitet: 426 gearbeitet, aber keine Exposition: 427

Tabelle 6.1: Übersicht über die Expositionen über alle Tätigkeiten und Jahre hinweg

In Abbildung 6.6 sind die Boxplots für die Exposition über alle Tätigkeiten und Jahre hinweg dargestellt.
Der Interquartilsabstand der Boxplots ist bei allen fünf Kategorien relativ breit, am stärksten fällt hier die Kategorie IRRPEAKS mit den breitesten Interquartilsabstand auf.
Ausreißer mit vergleichsweise hohen Expositionsdauern traten bei allen Expositionskategorien bis auf IRRPEAKS auf.

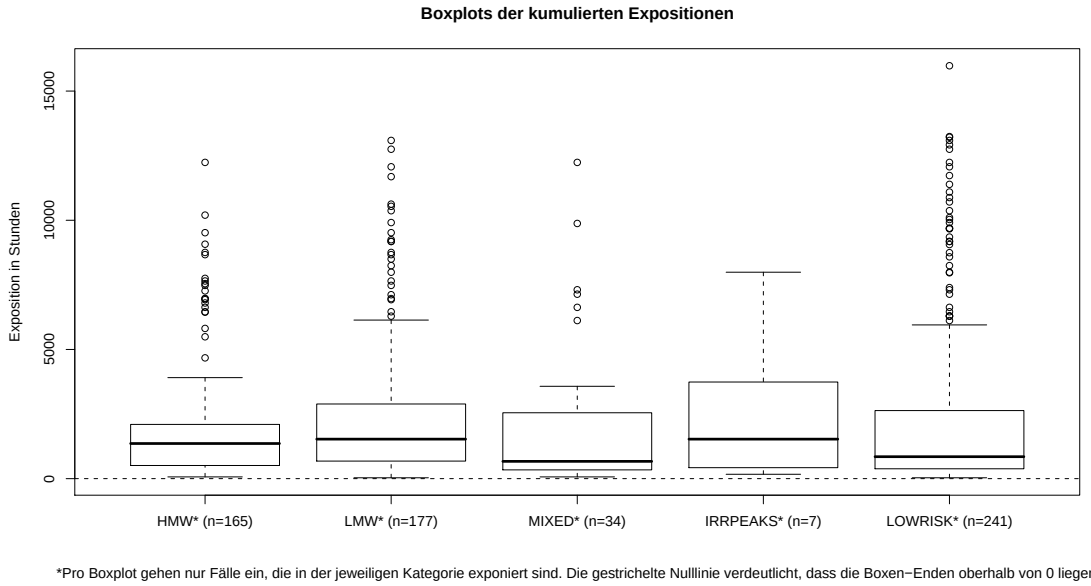


Abbildung 6.6: Boxplots der kumulierten Expositionen auf Basis der vollständigen Tätigkeitsangaben

Exposition im ersten Tätigkeitsjahr

Tabelle 6.2 stellt die Expositionen im ersten Tätigkeitsjahr dar.
Das Auftreten der Expositionen im ersten Tätigkeitsjahr war sehr ähnlich zum Auftreten der Expositionen über alle Tätigkeiten und Jahre hinweg. Es waren jedoch nur rund 2/3 der Probanden, die während ihres gesamten bisherigen Arbeitslebens einer Exposition ausgesetzt waren, bereits im ersten Tätigkeitsjahr exponiert. Während des ersten Tätigkeitsjahrs traten ebenfalls am häufigsten Expositionen in den Kategorien LOWRISK, LMW und HMW auf. Relativ selten waren die Probanden in den Kategorien MIXED und IRRPEAKS exponiert.

Beim Vergleich der Mediane der Expositionen in den verschiedenen Kategorien lag die mit Abstand längste Exposition in der Kategorie IRRPEAKS vor. Die Mediane der Expositionen in den anderen vier Kategorien lagen deutlich darunter.

Exposition im 1. Tätigkeitsjahr	**Exposition vorhanden**	**Keine Exposition vorhanden**
HMW	Anzahl Fälle: 114 Median: 697 Stunden Range in Stunden: [68,2295]	Anzahl Fälle ohne Exposition: 980 nie gearbeitet: 426 gearbeitet, aber keine Exposition: 554
LMW	Anzahl Fälle: 123 Median: 850 Stunden Range in Stunden: [34,3672]	Anzahl Fälle ohne Exposition: 971 nie gearbeitet: 426 gearbeitet, aber keine Exposition: 545
MIXED	Anzahl Fälle: 23 Median: 574 Stunden Range in Stunden: [68,2550]	Anzahl Fälle ohne Exposition: 1071 nie gearbeitet: 426 gearbeitet, aber keine Exposition: 645
IRRPEAKS	Anzahl Fälle: 4 Median: 1071 Stunden Range in Stunden: [170,2040]	Anzahl Fälle ohne Exposition: 1090 nie gearbeitet: 426 gearbeitet, aber keine Exposition: 664
LOWRISK	Anzahl Fälle: 176 Median: 510 Stunden Range in Stunden: [34,3672]	Anzahl Fälle ohne Exposition: 918 nie gearbeitet: 426 gearbeitet, aber keine Exposition: 492

Tabelle 6.2: Übersicht über die Expositionen im ersten Tätigkeitsjahr

In Abbildung 6.7 sind die Boxplots für die Exposition im ersten Tätigkeitsjahr dargestellt.
Zunächst fällt auf, dass die Interquartilsabstände hier deutlich schmaler sind als bei der Exposition über alle Tätigkeiten und Jahre hinweg. Der Interquartilsabstand des Boxplots für die Kategorie IRRPEAKS ist hier am schmälsten.
Sehr wenige Ausreißer mit vergleichsweise hohen Expositionsdauern lagen ausschließlich bei der LOWRISK-Exposition vor.

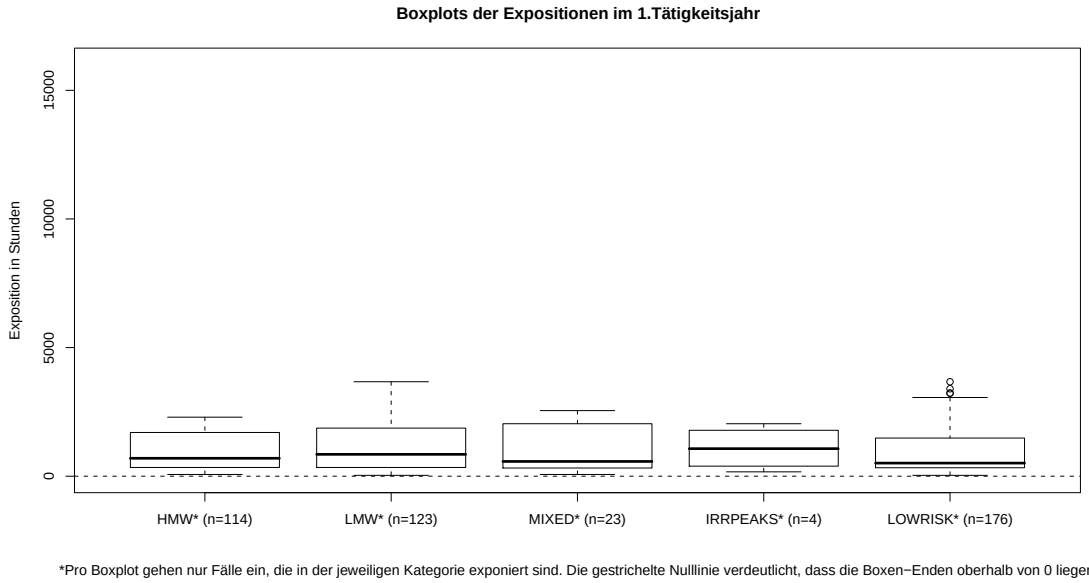


Abbildung 6.7: Boxplots der Expositionen im ersten Tätigkeitsjahr auf Basis der vollständigen Tätigkeitsangaben

Exposition in der ersten Tätigkeit

Tabelle 6.3 zeigt den Median, das Minimum und das Maximum der berechneten Expositionen während der ersten Tätigkeit.
60% der Personen, die während ihres gesamten bisherigen Arbeitslebens einer Exposition ausgesetzt waren, waren bereits während ihrer ersten Tätigkeit exponiert. Am häufigsten trat erneut eine Exposition in der Kategorie LOWRISK auf, gefolgt von den Kategorien LMW und HMW. Betrachtet man die Mediane der Expositionen in den verschiedenen Kategorien, so lag die längste Belastung in den Kategorien LMW und IRRPEAKS vor.

Exposition der 1. Tätigkeit	**Exposition vorhanden**	**Keine Exposition vorhanden**
HMW	Anzahl Fälle: 103 Median: 893 Stunden Range in Stunden: [68,12240]	Anzahl Fälle ohne Exposition: 991 nie gearbeitet: 426 gearbeitet, aber keine Exposition: 565
LMW	Anzahl Fälle: 108 Median: 1077 Stunden Range in Stunden: [34,13090]	Anzahl Fälle ohne Exposition: 986 nie gearbeitet: 426 gearbeitet, aber keine Exposition: 560
MIXED	Anzahl Fälle: 18 Median: 510 Stunden Range in Stunden: [68,10710]	Anzahl Fälle ohne Exposition: 1076 nie gearbeitet: 426 gearbeitet, aber keine Exposition: 650
IRRPEAKS	Anzahl Fälle: 4 Median: 1071 Stunden Range in Stunden: [170,3570]	Anzahl Fälle ohne Exposition: 1090 nie gearbeitet: 426 gearbeitet, aber keine Exposition: 664
LOWRISK	Anzahl Fälle: 163 Median: 612 Stunden Range in Stunden: [34,15980]	Anzahl Fälle ohne Exposition: 931 nie gearbeitet: 426 gearbeitet, aber keine Exposition: 505

Tabelle 6.3: Übersicht über die Expositionen in der ersten Tätigkeit

In Abbildung 6.8 sind die Boxplots für die Exposition in der ersten Tätigkeit dargestellt. Die Interquartilsabstände der hier dargestellten Boxplots sind deutlich schmaler als die Interquartilsabstände der Boxplots für die Exposition über alle Tätigkeiten und Jahre hinweg, allerdings etwas breiter als die Interquartilsabstände der Boxplots für die Exposition im ersten Tätigkeitsjahr. Ausreißer mit vergleichsweise langen Expositionsdauern traten hauptsächlich in den Kategorie LOWRISK, HMW und LMW auf, vereinzelt auch in der Kategorie MIXED.

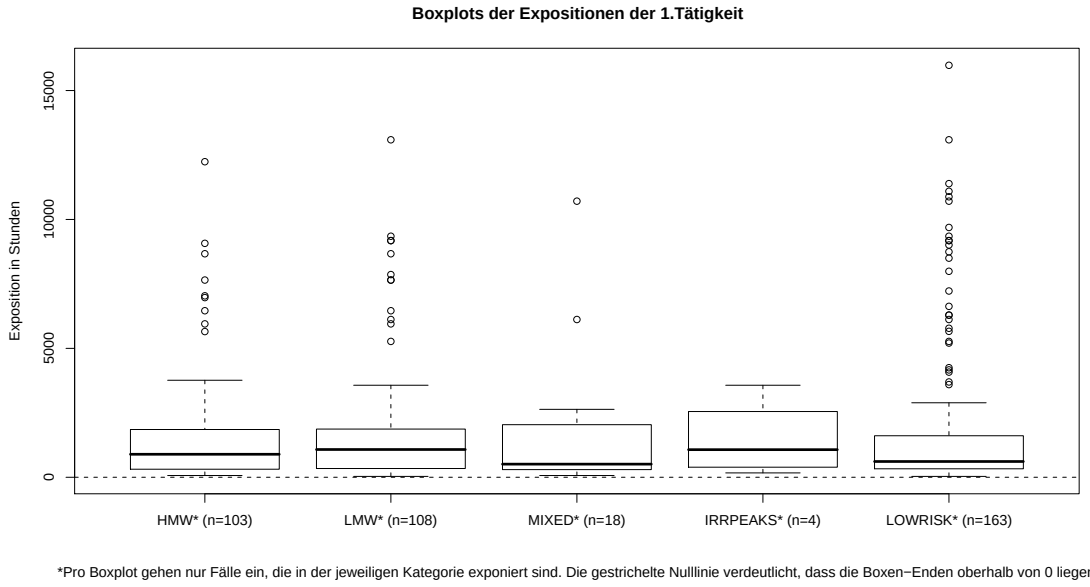


Abbildung 6.8: Boxplots der Expositionen in der ersten Tätigkeit auf Basis der vollständigen Tätigkeitsangaben

Betrachtung der binären Expositionsvariablen

In Abbildung 6.9 ist für die drei Expositionsarten (Kumulierte Exposition, Exposition im ersten Tätigkeitsjahr, Exposition in der ersten Tätigkeit) abgebildet, wie viele Probanden jeweils in welcher Kategorie exponiert waren.
Im Vergleich zu den anderen Expositionskategorien waren in der Kategorie LOWRISK die meisten Probanden exponiert, gefolgt von den Kategorien LMW und HMW. In den beiden Kategorien MIXED und IRRPEAKS lag nur bei relativ wenigen Probanden eine Exposition vor.

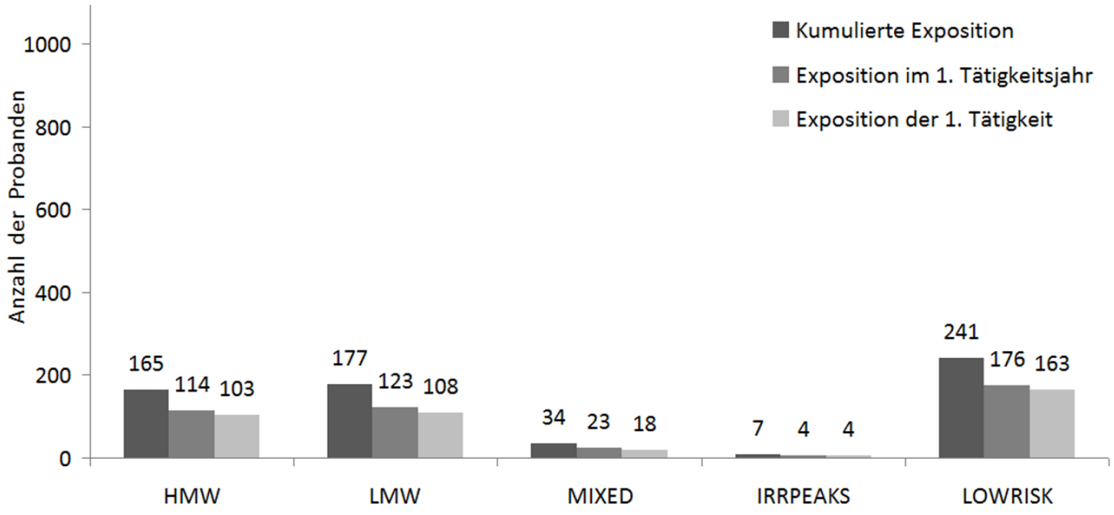


Abbildung 6.9: Binäre Expositionsvariblen

KAPITEL 7

Logistische Regression

Im Rahmen dieser Bachelorarbeit wurden Regressionsmodelle für die beiden Zielgrößen Asthma in SOLAR II und allergische Rhinitis in SOLAR II erstellt. Diese beiden Zielgrößen sind binäre Variablen und wie folgt kodiert:

$$s2CURASTHV = \begin{cases} 1, & Asthma\ in\ SOLAR\ II \\ 0, & Kein\ Asthma\ in\ SOLAR\ II \end{cases}$$

$$s2CURHAYV = \begin{cases} 1, & allergischeRhinitis\ in\ SOLAR\ II \\ 0, & Kein\ allergischeRhinitis\ in\ SOLAR\ II \end{cases}$$

Weil die Zielgrößen binär sind, ist ein logistisches Regressionsmodell geeignet.

7.1 Modellannahmen

Das Ziel einer logistischen Regressionsanalyse ist die Modellierung und Analyse der bedingten Wahrscheinlichkeit

$$\pi_i = P(y_i = 1) = P(y_i = 1|x_{i1}, ..., x_{ip}) = E(Y_i = 1|x_{i1}, ..., x_{ip})$$

in Abhängigkeit von den Kovariablen.
Das Modell kann durch folgende Formen dargestellt werden:

$$\pi_i = \frac{\exp(\beta_0 + \beta_1 x_{i1} + ... + \beta_p x_{ip})}{1 + \exp(\beta_0 + \beta_1 x_{i1} + ... + \beta_p x_{ip})} \tag{7.1}$$

$$\log\left(\frac{\pi_i}{1 - \pi_i}\right) = \beta_0 + \beta_1 x_{i1} + ... + \beta_p x_{ip} \tag{7.2}$$

$$\frac{\pi_i}{1-\pi_i} = \exp(\beta_0 + \beta_1 x_{i1} + ... + \beta_p x_{ip}) \tag{7.3}$$

Dabei wird in (7.1) der bedingte Erwartungswert $E(y_i|x_i)$ durch π_i modelliert. Durch Umformung erhält man die Formen in den Gleichungen (7.2) und (7.3), wobei durch $\frac{\pi}{1-\pi}$ die Chancen ("Odds") und durch $\log\left(\frac{\pi_i}{1-\pi_i}\right)$ die logarithmierten Chancen ("Logits") dargestellt werden.

7.2 Parameterschätzung

Die Schätzung der Parameter im logistischen Regressionsmodell erfolgt mit der Maximum-Likelihood-Methode.
Das Maximum-Likelihood-Prinzip besagt: Wähle zu den Realisationen denjenigen Parameter als Parameterschätzung, für den die Likelihood maximal ist. Man wählt also denjenigen Parameter, für den die Wahrscheinlichkeit, dass gerade diese Werte auftreten, maximal wird.
Das Maximum der Likelihood kann bestimmt werden durch Ableiten, Null setzen und Auflösen des resultierenden Gleichungssystems nach dem Parametervektor β. Aus technischen Gründen verwendet man üblicherweise statt der Likelihood die Log-Likelihood, die durch Logarithmieren der Likelihood gebildet wird. Durch Ableiten der Log-Likelihood nach β erhält man die Scorefunktion. Durch Nullsetzen der Scorefunktion ergibt sich die ML-Gleichung, die üblicherweise iterativ durch den Fisher-Scoring-Algorithmus gelöst wird. Dieses Verfahren liefert dann den Maximum-Likelihood-Schätzer $\widehat{\beta}$ für β.

7.3 Parameterinterpretation

Interpretation der Logits

Die Interpretation der Parameter aufgrund der Gleichung (7.2) lautet: Bei Zunahme von x_{ij} um eine Einheit (von x_{ij} auf $x_{ij}+1$) verändern sich die Logits bei festgehaltenen restlichen Kovariablen additiv um β_j.

Interpretation der Chancen

Eine alternative, anschaulichere Interpretation lässt sich auf der Basis von Gleichung (7.3) durchführen: Nimmt x_{ij} um eine Einheit zu (von x_{ij} auf $x_{ij}+1$), so verändert sich die Chance bei festgehaltenen restlichen Kovariablen multiplikativ um $\exp(\beta_j)$. Die Chance vergrößert sich für $\beta_j > 0$, verkleinert sich für $\beta_j < 0$ und bleibt für $\beta_j = 0$ unverändert.

Interpretation der Odds Ratios

Als weitere Möglichkeit kann man auch die Odds Ratios (Chancenverhältnisse) interpretieren.
So kann man zum Beispiel das Odds Ratio von Rauchern gegenüber Nichtrauchern betrachten. Sei

$$x_R = \begin{cases} 1, & Raucher\ in\ SOLAR\ I \\ 0, & Nichtraucher\ in\ SOLAR\ I \end{cases}$$

die Kovariable des logistischen Modells, die angibt, ob der Proband in SOLAR I Raucher oder Nichtraucher war, und β_R der zugehörige Parameter. Dann berechnet sich die Chance für Raucher folgendermaßen:

$$\gamma(x_R = 1) = \frac{\pi(x_R=1)}{1-\pi(x_R=1)}.$$

Analog kann auch die Chance für Nichtraucher berechnet werden:

$$\gamma(x_R = 0) = \frac{\pi(x_R=0)}{1-\pi(x_R=0)}.$$

Das Odds-Ratio (OR) ist das Chancenverhältnis, also der Quotient der beiden Chancen:

$$OR = \frac{\gamma(x_R=1)}{\gamma(x_R=0)} = \frac{\frac{\pi(x_R=1)}{1-\pi(x_R=1)}}{\frac{\pi(x_R=0)}{1-\pi(x_R=0)}} = \frac{\exp(\beta_0)\cdot\exp(\beta_R)}{\exp(\beta_0)} = \exp(\beta_R).$$

Wenn man nun das Odds Ratio von Rauchern gegenüber Nichtrauchern beispielsweise im Zusammenhang mit der Zielgröße Asthma in SOLAR II bei festgehaltenen restlichen Kovariablenwerten interpretieren will, so gilt: Ist das Odds Ratio gleich 1, so sind die Chancen in SOLAR II Asthma zu haben für Raucher und Nichtraucher gleich. Ist das Odds Ratio größer als 1, so gilt: Raucher haben im Gegensatz zu Nichtrauchern ein erhöhtes Asthmarisiko in SOLAR II. Ist das Odds Ratio kleiner als 1, so gilt: Raucher haben im Gegensatz zu Nichtrauchern ein erniedrigtes Asthmarisiko in SOLAR II.

Ein 95%-Konfidenzintervall für das Odds-Ratio kann wie folgt berechnet werden:

$$95\%\text{-KI} = [\exp(\widehat{\beta} - 1.96\ \widehat{\sigma}),\ \exp(\widehat{\beta} + 1.96\ \widehat{\sigma})]$$

7.4 Likelihood-Quotienten-Test

Mit dem Likelihood-Quotienten-Test können Hypothesentests der Form

$$H_0 : C\beta = d \quad vs. \quad H_1 : C\beta \neq d$$

$(mit \quad rg(C) = r \leq p, wobei \quad p : Parameteranzahl)$ durchgeführt werden.
Der Likelihood-Quotienten-Test kann deswegen im Rahmen der Regressionsanalyse zum Vergleich von hierarchischen Modellen verwendet werden. Mit ihm kann geprüft werden, ob die Aufnahme einer oder mehrerer zusätzlicher Kovariablen in ein Regressionsmodell zu einer Modellverbesserung führt, oder ob das Modell auch ohne die zusätzliche(n) Kovariable(n) ausreichend informativ ist. Dabei lautet die Nullhypothese: Die Aufname der zusätzlichen Kovariable(n) führt zu keiner Modellverbesserung. Wird die Nullhypothese abgelehnt, so wird davon ausgegangen, dass die Aufnahme der zusätzlichen Kovariable(n) in das Regressionsmodell zu einer Modellverbesserung führt und deswegen das größere Modell (unrestringiertes Modell), das die zusätzliche(n) Kovariable(n) enthält, gegenüber dem kleineren Untermodell (restringiertes Modell) bevorzugt werden sollte. Das bedeutet, der (die) Parameterschätzer der neu aufgenommenen Kovariable(n) unterscheidet (unterscheiden) sich signifikikant von 0.
"Die Likelihood-Quotienten-Statistik

$$lq = -2\{l(\widetilde{\beta}) - l(\widehat{\beta})\}$$

misst die Abweichung zwischen dem unrestringierten Maximum $l(\widehat{\beta})$ und dem unter H_0 restringierten Maximum $l(\widetilde{\beta})$, wobei $\widetilde{\beta}$ ML-Schätzer unter der Gleichungsrestriktion $C\beta = d$ ist." [Fahrmeir et al. 2007]

7.5 Variablenselektion und Modellwahl: AIC-Kriterium

AIC-Kriterium (Akaikes Informationskriterium)

Beim Vergleich von Modellen mit verschiedenen Parametern muss ein Kompromiss zwischen einer guten Datenanpassung durch Modellkomplexität (hohe Parameterzahl) und Modelleinfachheit (geringe Parameterzahl) getroffen werden. Durch Anwendung von Modellwahlkriterien soll ein solcher Kompromiss gefunden werden. Das wohl bekannteste Kriterium für die Modellwahl in der logistischen Regression ist das AIC-Kriterium:

$$AIC = -2l(\widehat{\beta}) + 2p.$$

In den ersten Term geht die Log-Likelihood mit dem ML-Schätzer $\widehat{\beta}$ ein. Dieser Term gibt also in etwa an, wie gut das Modell den vorliegenden Daten angepasst ist. Der Term

$2p$ (p ist die Parameteranzahl) bestraft die Anzahl der Parameter in einem zu komplexen Modell. Bei der Wahl zwischen verschiedenen Modellen wird das Modell mit dem kleinsten AIC-Wert bevorzugt.

Anwendung des AIC-Kriteriums

Das AIC-Kriterium kann zur Selektion der "besten Modellen" aus einer Reihe zur Verfügung stehender Modelle verwendet werden.
Durch substanzwissenschaftliche Überlegungen sollte jedoch zu Beginn der Modellwahl eine Auswahl potentieller Modelle getroffen werden. In dieser Arbeit wurde so zum Beispiel festgelegt, dass die beiden Variablen "Geschlecht" und "sozioökonomischer Status" fest ins Modell aufgenommen werden sollen, da man annimmt, dass diese beiden Variablen im Zusammenhang mit den beiden Zielgrößen Asthma und allergische Rhinitis stehen.
Eine Variante um Modellwahl zu betreiben wäre, alle möglichen, in Frage kommenden Modelle zu berechnen und das Modell, das zum Beispiel den kleinsten AIC-Wert aufweist, auszuwählen. Da die Berechnung aller möglichen Modelle nicht immer durchführbar ist, da zum Beispiel die Zahl der möglichen Einflussgrößen sehr groß ist (in dieser Bachelorarbeit gäbe es bei 20 in Frage kommenden Confounder-Variablen schon 1.048.576 ($= 2^{20}$) mögliche Modelle), werden oft Selektionsverfahren (z.B. auf Basis des AIC-Kriteriums) angewandt, um auch ohne die Berechnung aller möglichen Modelle zu einem sehr guten Modell zu gelangen. Im Folgenden werden drei Selektionsverfahren auf Basis des AIC-Kriteriums vorgestellt.

- Vorwärts-Selektion (Forward-Selection)
  Ausgehend von einem minimalen Modell (kleinste Anzahl an Einflussgrößen) wird in jedem Schritt des Selektionsverfahrens eine weitere Einflussgröße ins Modell aufgenommen. Es wird diejenige Einflussgröße aufgenommen, welche die größte Reduktion des AIC liefert. Das wird so lange durchgeführt, bis keine Reduktion des AIC mehr möglich ist.

- Rückwärts-Selektion (Backward-Selection)
  Gestartet wird mit dem maximalen Modell (maximale Anzahl an Einflussgrößen). In jedem Schritt wird diejenige Einflussgröße aus dem Modell entfernt, welche die größte Reduktion des AIC liefert. Das wird so lange durchgeführt, bis keine Reduktion des AIC mehr möglich ist.

- Schrittweise Selektion (Stepwise-Selection)
  Dieses Verfahren ist eine Kombination aus Vorwärts- und Rückwärts-Selektion. Es kann in jedem Schritt sowohl eine Einflussgröße aufgenommen als auch entfernt

werden. Auch Einflussgrößen, die in einem vorangegangenen Schritt bereits aus dem Modell entfernt wurden, können später wieder mit aufgenommen werden und umgekehrt.

Obwohl diese drei Verfahren nicht zum besten Modell im Sinne der Modellwahlkriterien führen, da nicht alle möglichen Modelle berechnet werden, führen sie in der Regel zu einem sehr guten Modell.

Die hier ausgeführte Theorie zu Akaikes Informationskriterium und zur Anwendung des AIC-Kriteriums geht größtenteils auf [Fahrmeir et al. 2007] zurück.

7.6 GAM (Generalized Additive Model)

Die generalisierten additiven Modelle (GAM) sind eine Art Erweiterung der generalisierten linearen Modelle (GLM), zu denen das logistische Regressionsmodell gehört. In folgender Situation ist die Anwendung eines GAMs angebracht:
Es liegen für ein Regressionsmodell Kovariablen $x_{i1}, ..., x_{ik}$ vor. Der Einfluss dieser Kovariablen auf die Zielgröße y_i des Modells kann durch einen linearen Prädiktor modelliert werden. Zusätzlich liegen weitere, metrische Kovariablen $z_{i1}, ..., z_{iq}$ vor, von denen nicht bekannt ist, ob sie durch einen linearen Prädiktor modelliert werden können. Der Einfluss dieser weiteren Kovariablen soll also nichtparametrisch modelliert werden.
Die Modellgleichung des GAMs lautet:

$$\log\left(\frac{\pi_i}{1-\pi_i}\right) = \beta_0 + \beta_1 x_{i1} + ... + \beta_k x_{ik} + f_1(z_{i1}) + ... + f_q(z_{iq})$$

Die Funktionen $f_1(z_{i1}), ..., f_q(z_{iq})$ für die Kovariablen $z_{i1}, ..., z_{iq}$, die durch das GAM geschätzt werden, wirken additiv zusammen und werden als glatt vorausgesetzt.
Im Rahmen dieser Bachelorarbeit werden GAMs genutzt, um den Einfluss der metrischen Expositionsvariablen auf die jeweilige Zielgröße zu modellieren.
Die hier ausgeführte Theorie zu den generalisierten additiven Modellen geht größtenteils auf [Fahrmeir et al. 2007] zurück.

7.7 ROC-Analyse

In der Epidemiologie wird aufgrund von diagnostischen Tests, die positiv (T^+) oder negativ (T^-) ausfallen können, entweder angenommen, dass eine bestimmte Krankheit vorliegt (K) oder es wird angenommen, dass die Krankheit nicht vorliegt ($\overline{K}$). Es gibt in diesem Zusammenhang wichtige bedingte Wahrscheinlichkeiten, welche die Brauchbarkeit eines Tests zur Erkennung einer Krankheit angeben:

- **Sensitivität:** $P(T^+|K)$
  Wahrscheinlichkeit, dass ein Kranker ein positives Testergebnis hat
- **Spezifität:** $P(T^-|\overline{K})$
  Wahrscheinlichkeit, dass ein Gesunder ein negatives Testergebnis hat

Diese beiden Wahrscheinlichkeiten sollen bei einem guten Test möglichst groß sein.

Das Ergebnis eines solchen Tests ist oft eine kontinuierliche Messgröße. In dieser Bachelorarbeit werden zwei logistische Modelle für die Zielgrößen "Asthma in SOLAR II" und "Allergische Rhinitis in SOLAR II" gerechnet. Das Ergebnis dieser beiden logistischen Regressionsmodelle, die in diesem Zusammenhang als (diagnostische) Tests zu verstehen sind, sind Wahrscheinlichkeiten, die angeben, mit welcher Wahrscheinlichkeit ein Proband mit bestimmten Kovariablenwerten an Asthma bzw. allergischer Rhinitis erkrankt ist. Es gibt also keine "natürliche" Grenze zwischen "erkrankt" und "nicht erkrankt". Deswegen sind die geschätzte Sensitivität und Spezifität abhängig von der Festlegung eines Trennwertes ("cut off value"). Durch diesen Trennwert soll der Anteil falsch positiver und/oder falsch negativer Entscheidungen möglichst gering gehalten werden. Die Wahl des optimalen Trennwerts ist auch abhängig von den Risiken falscher Entscheidungen und substanzwissenschaftlichen Überlegungen.
Um einen optimalen Trennwert festzulegen, wird oftmals das ROC-Verfahren ("receiver operating characteristic") angewandt. Hierbei werden über den gesamten Definitionsbereich der Messgröße in diskreten Schritten möglichst viele Werte durchlaufen und an jedem Punkt die zugehörigen Sensitivitäten und Spezifitäten berechnet. Das Ergebnis dieses Verfahrens wird dann als ROC-Kurve graphisch dargestellt.
Bei einem optimalen Trennwert sollen insbesondere die Sensitivität und die Spezifität möglichst hoch liegen. Ein Test ist im Allgemeinen umso besser, je größer die Fläche unter der ROC-Kurve ist. Diese Fläche wird AUC ("area under the curve") genannt. Die AUC kann maximal 1 sein, was der Fall ist, wenn Spezifität und Sensitivität 100% betragen. Ist keine Trennung möglich, so ist die AUC 0.5, das heisst die ROC-Kurve verläuft entlang der Winkelhalbierenden.
Die Theorie zur ROC-Analyse in diesem Kapitel basiert größtenteils auf [Sachs und Hedderich 2006].

7.8 Logistische Regressionsmodelle für die Probanden mit vollständigen Tätigkeitsdaten

Auf Basis der Probanden, die vollständige Tätigkeitsangaben gemacht hatten (n=1.094), wurden nun zwei logistische Modelle angepasst. Die Zielgrößen für die logistischen Modelle sind "Allergische Rhinitis in SOLAR II" und "Asthma in SOLAR II". Sie wurden getrennt voneinander modelliert. Es wurde festgelegt, dass nur Haupteffekte und keine Interaktionen in die Modelle eingehen sollten.
Bevor die Modellwahl für das logistische Regressionsmodell für die Probanden mit vollständigen Tätigkeitsdaten ausführlich behandelt wird, wird das Vorgehen bei der Modellwahl in Abbildung 7.1 graphisch dargestellt.

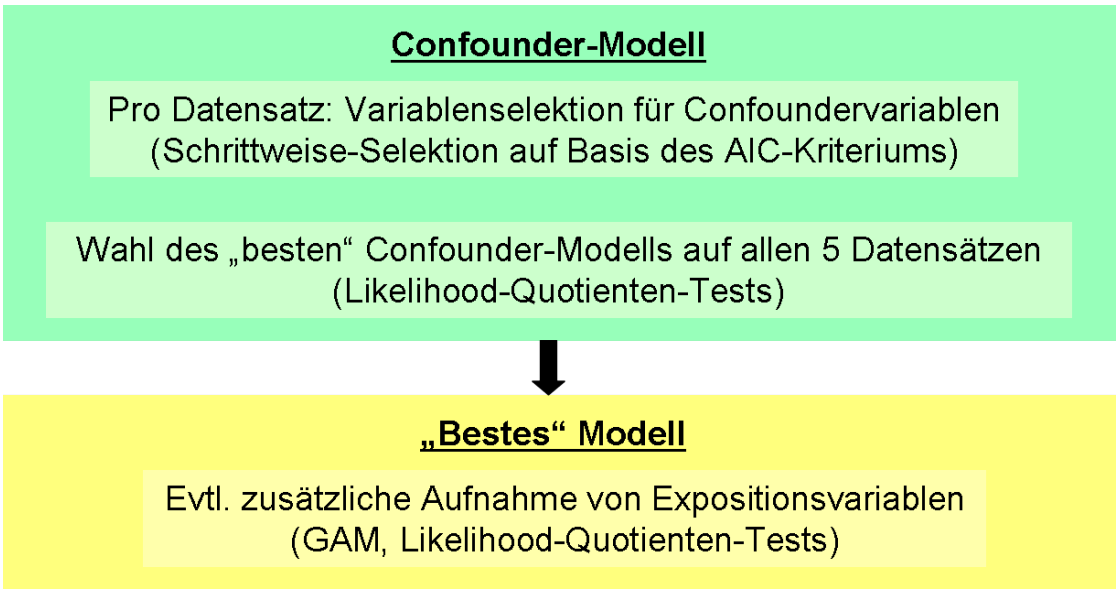


Abbildung 7.1: Vorgehen bei der Auswahl der logistischen Regressionsmodell für die Probanden mit vollständigen Tätigkeitsangaben

Die beiden Logit-Modelle wurden auf jedem der fünf Datensätze, welche jeweils imputierte Confoundervariablen und die vollständigen Tätigkeitsdaten enthalten, gefittet. Diese fünf Datensätze entstanden dadurch, dass der Datensatz, der die Confoundervariablen mit fehlende Werten enthielt, fünf mal imputiert wurde und somit fünf vervollständigte Datensätze resultierten. Diese fünf Datensätze wurden dann jeweils mit den Tätigkeitsangaben der Probanden mit vollständigen Tätigkeitsdaten zusammengefügt. Für jeden der so entstandenen Datensätze wurde zuerst ein Confounder-Modell gefittet, für das als mögliche Einflussgrößen die nachfolgend aufgeführten Variablen in Frage kamen.

7.8.1 Mögliche Einflussgrößen ("Confounder") für die logistischen Modelle

Variablen aus **ISAAC II**:

- Studienzentrum (Dresden/München)
- In Deutschland geboren (Ja/Nein)
- Atopie der Eltern (Ja/Nein)
- Anzahl Geschwister
- Gestillt (Ja/Nein)
- Neurodermitis (Ja/Nein)
- Allergische Rhinitis (Ja/Nein)
- Sozioökonomischer Status (Hoch/Niedrig)
- Passivrauch (Eltern: Raucher/Ex-Raucher/Nichtraucher)
- Asthma (Ja/Nein)

Variablen aus **SOLAR I**:

- Geschlecht (Männlich/Weiblich)
- Neurodermitis (Ja/Nein)
- Allergische Rhinitis (Ja/Nein)
- Passivrauch (Ja/Nein)
- Rauchverhalten (Raucher/Nichtraucher)
- Asthma (Ja/Nein)

Variablen aus **SOLAR II**:

- Passivrauch (Ja/Nein)
- Rauchverhalten (Raucher/Ex-Raucher/Nichtraucher)
- Schulbildung (Höhere/Niedrigere)

Aus **SOLAR I** und **SOLAR II** zusammengefasste Variable:

- Jemals gearbeitet (Ja/Nein)

Das Alter wurde nicht als mögliche Einflussgröße ("Confounder") gesehen, da sich alle Probanden in etwa im gleichen Alter befanden. Sie waren zum Zeitpunkt der SOLAR II-Studie alle im Alter von 21-23 Jahren.
Für das Modell mit der Zielgröße "Allergische Rhinitis in SOLAR II" wurden diejenigen Probanden, die in ISAAC II oder SOLAR I Asthma hatten, aus der Analyse ausgeschlossen. Das heisst, die Variablen "Asthma (ja/nein)" aus ISAAC II und SOLAR I gingen hier nicht als Kovariablen mit ein und die Datenbasis für das Modell wurde somit für dieses Modell auf 1.032 Probanden (im Gegensatz zu 1.094 Probanden im Modell mit der Zielgröße "Asthma in SOLAR II") reduziert.

Datenbasis der logistischen Modelle:
- ◇ Zielgröße Asthma in SOLAR II: n=1.094
- ◇ Zielgröße Allergische Rhinitis in SOLAR II: n=1.032
  (Kein Asthma in ISAAC II, SOLAR I)

7.8.2 Variablenselektion und Modellwahl

Da für die 20 in Frage kommenden Einflussgrößen für die Confounder-Modelle nicht alle möglichen Modelle berechnet und miteinander verglichen werden konnten, wurde auf Basis der fünf Datensätze (imputierte Confoundervariablen und vollständige Tätigkeitsdaten) zuerst eine Variablenselektion durchgeführt, um die für die Modelle relevanten Variablen auszuwählen. Die Variablenselektion wurde durch schrittweise Selektion (Stepwise-Selection) auf Basis des AIC-Kriteriums durchgeführt.
Für die schrittweise Selektion mussten jeweils ein minimales und ein maximales Modell angegeben werden. Das minimale Modell enthielt die Kovariablen "Geschlecht" und "sozioökonomischer Status", die auf jeden Fall als Counfounder mit ins Modell aufgenommen werden sollten, sowie einen Intercept.
Das Geschlecht sollte fest in das Modell mit aufgenommen werden, da man davon ausgehen kann, dass sich die Abläufe, die sich im Bezug auf eine Krankheit und die Körperfunktion im Allgemeinen abspielen und Faktoren die dazu beitragen (z.B. eventuelle Risikofaktoren), zwischen den Geschlechtern unterscheiden. Da diese Unterschiede gerade in der Pubertät in Erscheinung treten, ist es in der vorliegenden Kohortenstudie, in der die Pubertät der Probanden mitbetrachtet wird, wichtig, diese Variable als Einflussgröße ins Modell aufzunehmen. Vor allem in Bezug auf Asthmaerkrankungen haben verschiedene Studien gezeigt, dass sich das Auftreten von Asthma bei den Geschlechtern unterscheidet. Bei Kindern bis 16 Jahren leiden Jungen häufiger an Asthma als Mädchen.

Später im Jugendlichenalter (17-23 Jahre) ist es umgekehrt [ANDERSON et al. 1992].
Der sozioökonomische Status sollte fest in das Modell mit aufgenommen werden, da man davon ausgehen kann, dass es für die Gesundheit und Entwicklung eines Kindes einen Unterschied macht, ob es in einem Haushalt mit "niedrigerem" oder "höherem" sozioökonomischen Status aufwächst. Nach der Hygiene-Hypothese von David Strachan bedeutet das Aufwachsen in einem Haushalt mit "höherem" Status für ein Kind oft, dass es steriler aufwächst als in einem Haushalt mit "niedrigerem" Status, somit beispielsweise weniger Keimen ausgesetzt ist und auch weniger abgehärtet wird.
Das maximale Modell hatte neben einem Intercept alle oben aufgeführten möglichen Einflussgrößen als Kovariablen. Für das Modell mit der Zielgröße "Allergische Rhinitis in SOLAR II" wurden Probanden mit Asthma in ISAAC II oder SOLAR I ausgeschlossen und somit waren diese Kovariablen hier nicht Bestandteil des maximalen Modells.

Bei dem **Modell mit "Allergischer Rhinitis in SOLAR II" als Zielgröße** resultierten durch schrittweisen Selektion drei Confounder-Modelle, die als "bestes" Confounder-Modell in Frage kamen. Es gab auch hier deshalb mehrere Modelle zur Auswahl, weil pro Datensatz mit imputierten Confoundervariablen und vollständigen Tätigkeitsdaten die schrittweise Selektion durchgeführt wurde und somit unterschiedliche Modelle resultieren konnten (für jeden Datensatz ein Modell).
Die Einflussgrößen der aus der schrittweise Selektion resultierenden Confounder-Modelle auf den einzelnen Datensätzen sind in Tabelle 7.1 dargestellt.

Confoundervariablen imputiert mit	**Atopie der Eltern**	**Allerg.Rhinitis (ISAAC II)**	**Allerg.Rhinitis (SOLAR I)**	**Geschlecht**
AmeliaII (1)	+	+	+	+
AmeliaII (2)	+	+	+	+
AmeliaII (3)	+	+	+	+
Empir.Vert. (1)	+	+	+	+
Empir.Vert. (2)	+	+	+	+

Confoundervariablen imputiert mit	**Sozioökonom. Status**	**Als Säugling gestillt**	**Passivrauchen (SOLAR I)**
AmeliaII (1)	+	-	-
AmeliaII (2)	+	+	-
AmeliaII (3)	+	-	-
Empir.Vert. (1)	+	+	+
Empir.Vert. (2)	+	+	+

Tabelle 7.1: Einflussgrößen der Confounder-Modelle für Allergische Rhinitis in SOLAR II

Durch Anwendung der schrittweise Selektion erhielt man bei dem **Modell mit "Asthma in SOLAR II als Zielgröße"** zwei Confounder-Modelle, die als "bestes" Confounder-Modell in Frage kamen. Es gab deswegen mehrere Modelle zur Auswahl, weil pro Datensatz mit imputierten Confoundervariablen und vollständigen Tätigkeitsdaten die schrittweise Selektion durchgeführt wurde und somit unterschiedliche Modelle resultieren konnten (für jeden Datensatz ein Modell).
Die Einflussgrößen der aus der schrittweise Selektion resultierenden Modelle auf den einzelnen Datensätzen sind in Tabelle 7.2 dargestellt.

Confoundervariablen imputiert mit	**Asthma (ISAAC II)**	**Neurodermitis (SOLAR I)**	**Allerg.Rhinitis (SOLAR I)**	**Asthma (SOLAR I)**
AmeliaII (1)	+	+	+	+
AmeliaII (2)	+	+	+	+
AmeliaII (3)	+	+	+	+
Empir.Vert. (1)	+	+	+	+
Empir.Vert. (2)	+	+	+	+

Confoundervariablen imputiert mit	**Rauchen (SOLAR I)**	**Geschlecht**	**Sozioökonom. Status**	**Passivrauchen (SOLAR II)**	**Neurodermitis (ISAAC II)**
AmeliaII (1)	+	+	+	-	-
AmeliaII (2)	+	+	+	-	-
AmeliaII (3)	+	+	+	+	+
Empir.Vert. (1)	+	+	+	+	+
Empir.Vert. (2)	+	+	+	+	+

Tabelle 7.2: Einflussgrößen der Confounder-Modelle für Asthma in SOLAR II

Auswahl des "besten" Confounder-Modells

Um nun herauszufinden, welches Confoundermodell jeweils für alle fünf Datensätze am Besten passt, wurden Likelihood-Quotienten-Tests durchgeführt. Mit deren Hilfe wurden nun für beide Modelle die zur Auswahl stehenden "besten" Condounder-Modelle gegeneinander getestet.

Beim Modell für die **Zielgröße "Allergische Rhinitis in SOLAR II"** wurden drei Modelle gegeneinander getestet. Das minimale Modell enthielt die Kovariablen "Atopie der Eltern", "Allergische Rhinitis in ISAAC II", "Allergische Rhinitis in SOLAR I", "Geschlecht" und "sozioökonomischer Status". Das mittlere Modell enthielt zusätzlich zu den Kovariablen des minimalen Modells die Kovariable "Als Säugling gestillt". Das maximale Modell enthielt zusätzlich zu den Kovariablen des minimalen Modells die Kovariablen "Als Säugling gestillt" und "Passivrauchen in SOLAR I". Da das mittlere Modell auf

einem der fünf Datensätze eine Verbesserung gegenüber dem minimalen Modell brachte, das maximalen Modell jedoch auf keinem Datensatz eine weitere Verbesserung brachte, wurde das mittlere Modell als "bestes" Confounder-Modell für die Zielgröße "Allergische Rhinitis in SOLAR II" ausgewählt.

Beim Modell für die **Zielgröße "Asthma in SOLAR II"** wurde getestet, ob das Modell mit den zusätzlichen Variablen "Passivrauchen in SOLAR II" und "Neurodermitis in ISAAC II" eine Verbesserung gegenüber dem Modell bringt, das diese beiden Variablen nicht enthält, sondern nur die Einflussgrößen "Asthma in ISAAC II", "Neurodermitis in SOLAR I", "Allergische Rhinitis in SOLAR I", "Asthma in SOLAR I", "Rauchen in SOLAR I", "Geschlecht", "sozioökonomischer Status". Da dies auf allen fünf Datensätzen nicht der Fall war wurde das Modell ohne diese beiden Variablen als "bestes" Counfounder-Modell für die Zielgröße "Asthma in SOLAR II" ausgewählt.

Tabelle 7.3 gibt eine Übersicht über die Einflussgrößen in den "besten" Confounder-Modellen für "Asthma in SOLAR II" und "Allergische Rhinitis in SOLAR II".

Einflussgrößen der "besten" Confounder-Modelle	
Asthma in SOLAR II	Allergische Rhinitis in SOLAR II
Geschlecht	Geschlecht
Sozioökonomischer Status	Sozioökonomischer Status
Rauchen (SOLAR I)	Atopie der Eltern
Asthma (ISAAC II)	Allergische Rhinits (ISAAC II)
Asthma (SOLAR I)	Allergische Rhinitis (SOLAR I)
Neurodermitis (SOLAR I)	Als Säugling gestillt
Allerg.Rhinitis (SOLAR I)	

Tabelle 7.3: Einflussgrößen der "besten" Confounder-Modelle

Aufnahme von Expositionsvariablen in die Modelle

Als weitere Einflussgrößen konnten zusätzlich sogenannte Expositionsvariablen in das Modell eingehen, welche die asthmaspezifische Exposition des jeweiligen Probanden angeben. Folgende Expositionsvariablen kamen als weitere Einflussgrößen für die logistischen Regressionsmodelle in Frage:

- Kumulierte Exposition über alle Tätigkeiten und Jahre
- Binäre Exposition über alle Tätigkeiten und Jahre
- Kumulierte Exposition in der ersten ausgeübten Tätigkeit
- Binäre Exposition in der ersten ausgeübten Tätigkeit
- Kumulierte Exposition im ersten Tätigkeitsjahr
- Binäre Exposition im ersten Tätigkeitsjahr

Dabei sollte jede dieser sechs verschiedenen Expositionen nur einzeln in die Modelle eingehen, nicht gemeinsam mit einer anderen Exposition.

Um festlegen zu können, ob diese Expositionsvariablen als lineare Terme in die Modelle eingehen sollen oder zum Beispiel als quadratische Terme, wurden GAMs (Generalized additive models) gerechnet, in welche die metrischen Expositionsvariablen als glatte Effekte aufgenommen wurden, deren Einfluss auf die entsprechende Zielgröße modelliert werden sollte.
In Abbildung 7.2 sind beispielhaft für das Modell für die Zielgröße “Allergische Rhinitis in SOLAR II” die geschätzten Funktionen für die Expositionsvariablen, welche die Kumulierte Exposition über alle Tätigkeiten und Jahre angeben, abgebildet. Da der Wert der Expositionsvariable “IRRPEAKS-Exposition kumuliert” nur sehr selten größer war als 0 (nur in 7 Fällen), konnte für diese Variable keine Funktion geschätzt werden. Für die anderen vier Expositionsvariablen zeigt sich ein linearer Verlauf der Funktionen. Die Konfidenzintervalle werden für große Werte der Expositionsvaribien breiter, da es jeweils nur wenige Beobachtungen mit sehr großen Werten in den Expositionsvariablen gab.

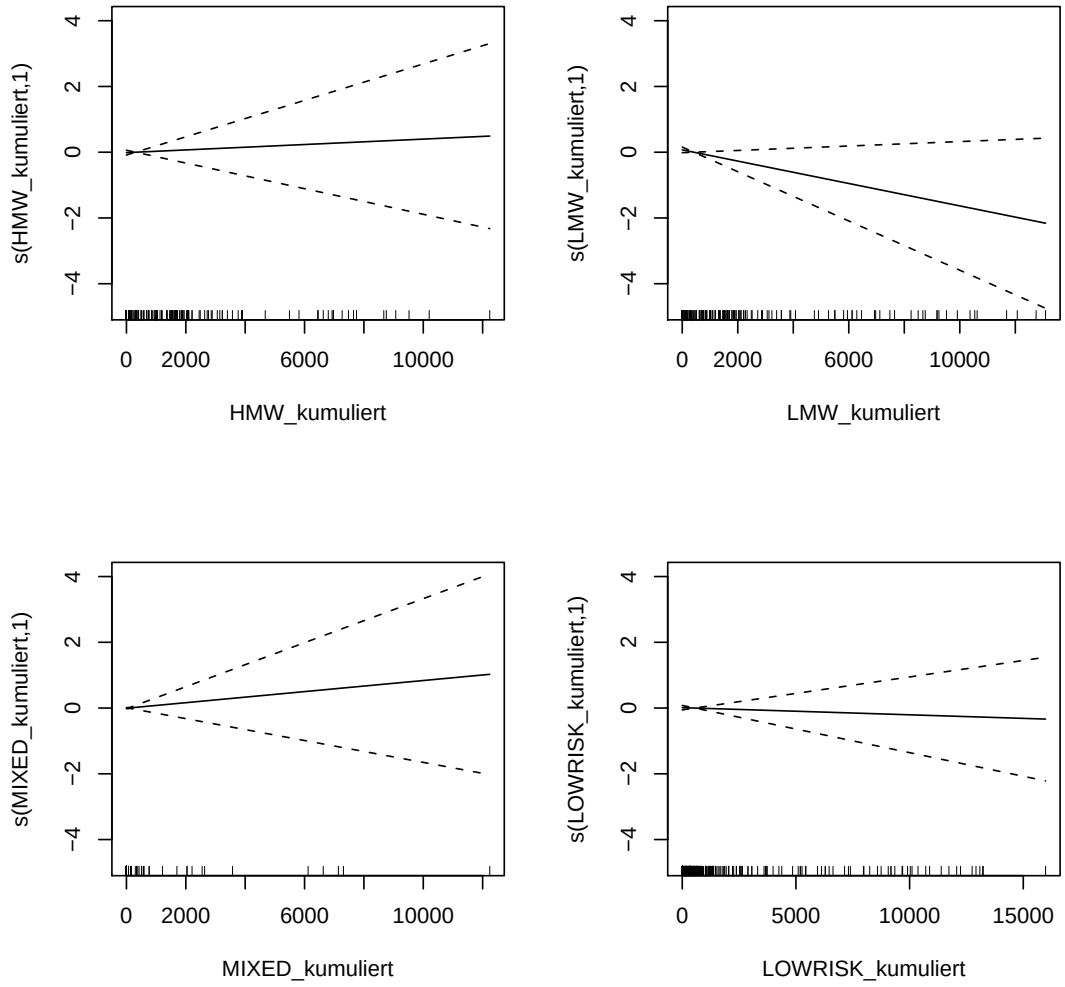


Abbildung 7.2: Geschätzte Funktionen für die Expositionsvariablen (Kumulierte Exposition über alle Tätigkeiten und Jahre)

Im **Modell für die Zielgröße "Allergische Rhinitis in SOLAR II"** konnten alle Expositionsvariablen als lineare Terme modelliert werden. Im **Modell für die Zielgröße "Asthma in SOLAR II"** wurde die Exposition im ersten Tätigkeitsjahr in der Kategorie HMW mit einem linearen und einem zusätzlichen quadratischen Term modelliert.

In einem weiteren Schritt wurde nun mithilfe von Likelihood-Quotienten-Tests überprüft, ob Expositionsvariablen ins Modell aufgenommen werden sollen oder nicht. Dazu wurde jeweils ein zweites Modell gebildet, das zusätzlich zu den im Confouner-Modell

enthaltenen Kovariablen die Expositionsvariablen für alle Kategorien (HMW, LMW, MIXED, IRRPEAKS, LOWRISK) als Einflussgrößen hatte und es wurde getestet, ob dieses Modell eine Verbesserung gegenüber dem Counfounder-Modell ohne die Expositionsvariablen bringt. Die Nullhypothese des Likelihood-Quotienten-Tests lautet hier also in etwa: Die zusätzliche Aufnahme der Expositionsvariablen führt zu keiner Modellverbesserung. Wird die Nullhypothese auf dem Signifikanzniveau von 5% abgelehnt, so nimmt man an, dass die Aufnahme der Expositionsvariablen zu einer Modellverbesserung führt.

Beim **Modell für die Zielgröße "Allergische Rhinitis in SOLAR II"** brachte die Aufnahme von Expositionsvariablen in keinem Fall eine Modellverbesserung, d.h., die Nullhypothese des Likelihood-Quotienten-Tests wurde in keinem Fall abgelehnt. Die p-Werte der durchgeführten Likelihood-Quotienten-Tests auf den verschiedenen Datensätzen sind der Tabelle 7.4 zu entnehmen.

Likelihood-Quotienten-Test Confoundermodell vs. Modell inklusive	**p-Wert** Datensatz 1	**p-Wert** Datensatz 2	**p-Wert** Datensatz 3	**p-Wert** Datensatz 4	**p-Wert** Datensatz 5
kumulierte Expositionen	0,45	0,43	0,45	0,44	0,45
binäre Expositionen	0,17	0,16	0,17	0,16	0,17
Expositionen des 1. Tätigkeitsjahres	0,59	0,57	0,59	0,59	0,59
binäre Expositionen des 1. Jahres	0,26	0,26	0,26	0,25	0,26
Expositionen der 1. Tätigkeit	0,27	0,25	0,28	0,27	0,27
binäre Expositionen der 1. Tätigkeit	0,67	0,66	0,67	0,66	0,67

Tabelle 7.4: Übersicht über die p-Werte der durchgeführten Likelihood-Quotienten-Tests - Modell für Allergische Rhinitis in SOLAR II

Aus inhaltlichen Gründen wurden jedoch die Expositionsvariablen für die binäre Exposition über alle Tätigkeiten und Jahre in das Modell mit aufgenommen, da bei der Aufnahme dieser Expositionsvariablen im Gegensatz zu den anderen Expositionsvariablen das AIC des logistischen Modells am kleinsten war. Hier wurden deswegen für das insgesamt "beste" Modell zusätzlich noch die Expositionsvariablen für die binäre Exposition über alle Tätigkeiten und Jahre in das "beste" Confounder-Modell aufgenommen. Die AIC-Werte der unterschiedlichen Modelle sind Tabelle 7.5 zu entnehmen.

Modell	**AIC** Datensatz 1	**AIC** Datensatz 2	**AIC** Datensatz 3	**AIC** Datensatz 4	**AIC** Datensatz 5
Confoundermodell (ohne Expositionsvariablen)	592,67	588,85	591,58	590,83	591,65
Confoundermodell inklusive Variablen für die kumulierte Exposition	597,97	593,95	596,87	596,06	596,93
Confoundermodell inklusive Variablen für die binäre Exposition	**594,96**	**590,99**	**593,75**	**592,85**	**593,92**
Confoundermodell inklusive Variablen für die Exposition des 1. Tätigkeitsjahres	598,96	595,03	597,88	597,08	597,95
Confoundermodell inklusive Variablen für die binäre Exposition des 1. Jahres	596,21	592,30	595,07	594,19	595,18
Confoundermodell inklusive Variablen für die Exposition der 1. Tätigkeit	596,31	592,22	595,26	594,46	595,31
Confoundermodell inklusive Variablen für die binäre Exposition der 1. Tätigkeit	599,47	595,62	598,38	597,54	598,45

Tabelle 7.5: Übersicht über die AIC-Werte der unterschiedlichen Modelle - Modell für Allergische Rhinitis in SOLAR II

Beim **Modell für die Zielgröße "Asthma in SOLAR II"** wurde für die Aufnahme der Expositionsvariablen für die kumulierte Exposition über alle Tätigkeiten und Jahre die Nullhypothese abgelehnt, d.h., man konnte davon ausgehen, dass die Aufnahme dieser Expositionsvariablen eine Verbesserung gegenüber dem Confounder-Modell bringt, wenn sie als lineare Terme aufgenommen werden. Die p-Werte der durchgeführten Likelihood-Quotienten-Tests auf den verschiedenen Datensätzen sind der Tabelle 7.6 zu entnehmen.

Likelihood-Quotienten-Test Confoundermodell vs. Modell inklusive	**p-Wert** Datensatz 1	**p-Wert** Datensatz 2	**p-Wert** Datensatz 3	**p-Wert** Datensatz 4	**p-Wert** Datensatz 5
kumulierte Expositionen	**0,04**	**0,04**	**0,04**	**0,04**	**0,04**
binäre Expositionen	0,38	0,38	0,38	0,39	0,38
Expositionen des 1. Tätigkeitsjahres	0,32	0,31	0,31	0,31	0,31
Expositionen des 1. Jahres (inkl. HMW als quadratischer Term)	0,22	0,21	0,21	0,21	0,21
binäre Expositionen des 1. Jahres	0,90	0,90	0,90	0,90	0,90
Expositionen der 1. Tätigkeit	0,15	0,15	0,15	0,15	0,15
binäre Expositionen der 1. Tätigkeit	0,79	0,79	0,79	0,79	0,80

Tabelle 7.6: Übersicht über die p-Werte der durchgeführten Likelihood-Quotienten-Tests - Modell für Asthma in SOLAR II

Hier wurden deswegen für das insgesamt "beste" Modell in das "beste" Confounder-Modell zusätzlich noch die Expositionsvariablen für die gesamte Exposition über alle Tätigkeiten und Jahre hinweg aufgenommen.

Tabelle 7.7 gibt eine Übersicht über die Einflussgrößen der insgesamt "besten" Modelle für "Asthma in SOLAR II" und "Allergische Rhinitis in SOLAR II".

Einflussgrößen der "besten" Modelle	
Asthma in SOLAR II	Allergische Rhinitis in SOLAR II
Geschlecht	Geschlecht
Sozioökonomischer Status	Sozioökonomischer Status
Rauchen (SOLAR I)	Atopie der Eltern
Asthma (ISAAC II)	Allergische Rhinits (ISAAC II)
Asthma (SOLAR I)	Allergische Rhinitis (SOLAR I)
Neurodermitis (SOLAR I)	Als Säugling gestillt
Allerg.Rhinitis (SOLAR I)	
HMW-Exposition kumuliert	HMW-Exposition binär
LMW-Exposition kumuliert	LMW-Exposition binär
MIXED-Exposition kumuliert	MIXED-Exposition binär
IRRPEAKS-Exposition kumuliert	IRRPEAKS-Exposition binär
LOWRISK-Exposition kumuliert	LOWRISK-Exposition binär

Tabelle 7.7: Einflussgrößen der "besten" Modelle

7.8.3 ROC-Analyse für die "besten" Modelle

Bestes Modell für die Zielgröße "Allergische Rhinitis in SOLAR II"

Für das "beste Modell" für die Zielgröße "Allergische Rhinitis in SOLAR II" wurde auf allen fünf Datensätzen eine ROC-Kurve erstellt. Abbildung 7.3 zeigt die ROC-Kurve für einen Datensatz, in dem die Confoundervariablen mit AMELIA II imputiert wurden. Auf den fünf Datensätzen beträgt die minimale AUC 0.837 und die maximale AUC 0.840.

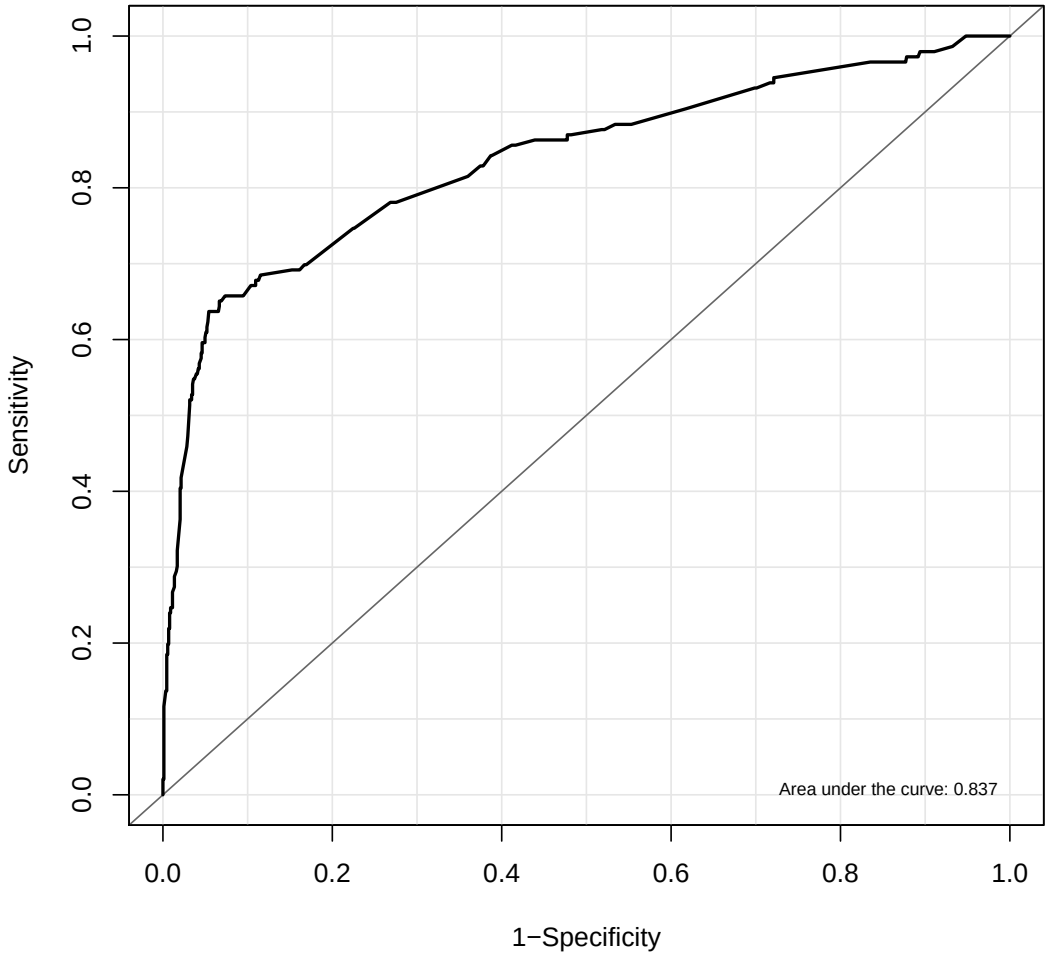


Abbildung 7.3: ROC-Kurve für das Modell mit der Zielgröße Allergische Rhinitis in SOLAR II

Bestes Modell für die Zielgröße "Asthma in SOLAR II"

Für das "beste Modell" für die Zielgröße "Asthma in SOLAR II" wurde auf allen fünf Datensätzen eine ROC-Kurve erstellt. Abbildung 7.4 zeigt die ROC-Kurve für einen Datensatz, in dem die Confoundervariablen mit AMELIA II imputiert wurden. Auf den fünf Datensätzen beträgt die minimale AUC 0.882 und die maximale AUC 0.887.

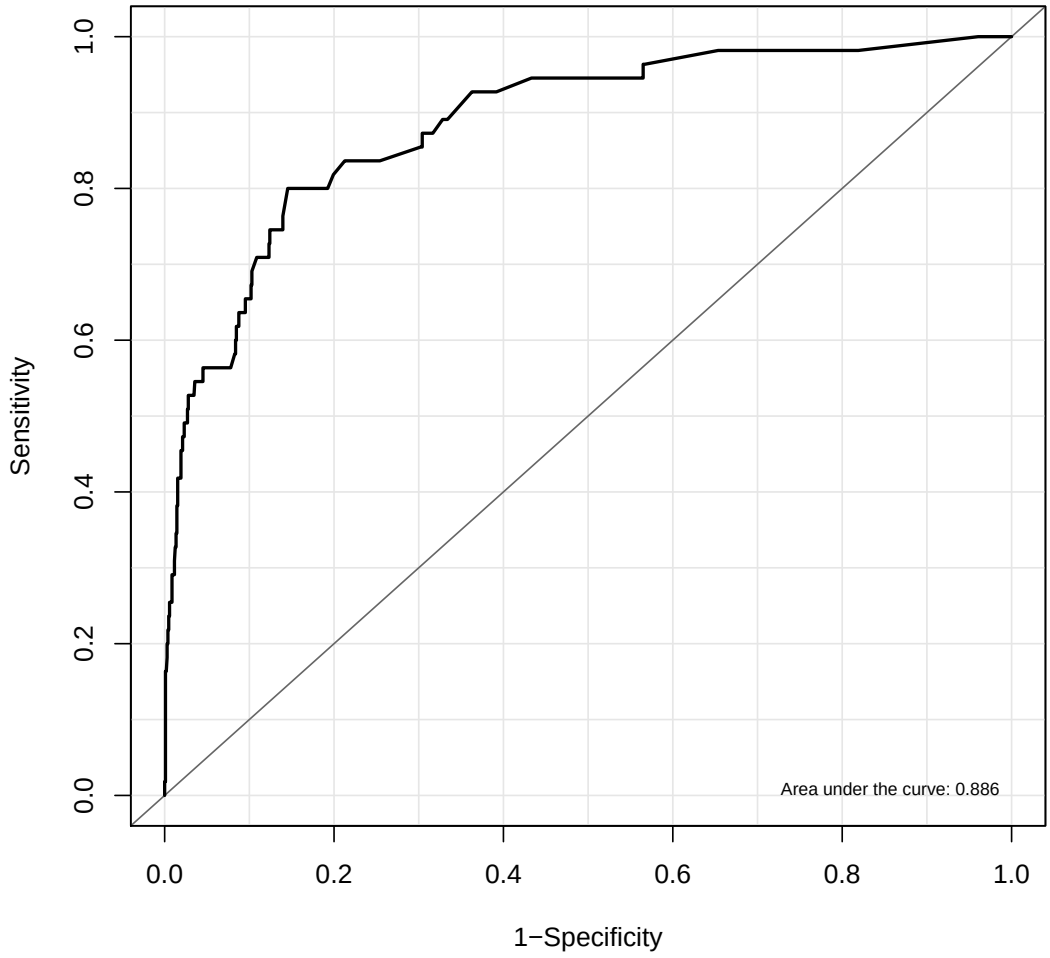
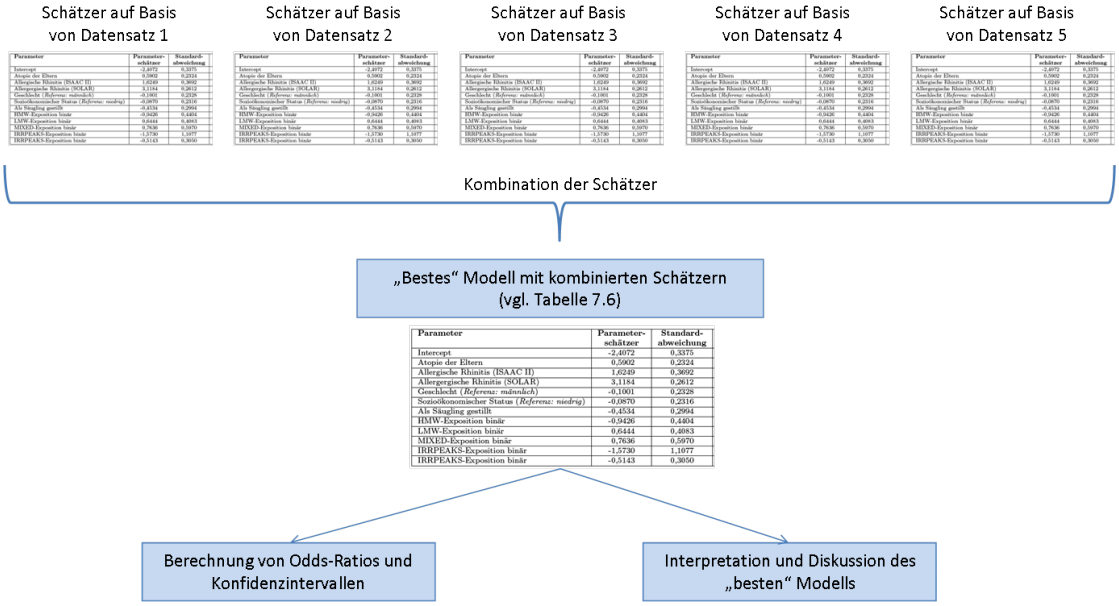


Abbildung 7.4: ROC-Kurve für das Modell mit der Zielgröße Asthma in SOLAR II

7.8.4 Schätzer kombinieren

Das “beste” Modell wurde nun jeweils auf allen fünf Datensätzen (imputierte Confoundervariablen und vollständige Tätigkeitsdaten) gerechnet. Die resultierenden Parameterschätzer dieser fünf Modelle wurden anschließend mit den in Kapitel 3 vorgestellten Formeln zur Kombination von Schätzern kombiniert.
In Abbildung 7.5 wird das Vorgehen bei der Kombination der Schätzer dargestellt.



Parameter	Parameter-schätzer	Standard-abweichung
Intercept	-2,4072	0,3375
Atopie der Eltern	0,5902	0,2324
Allergische Rhinitis (ISAAC II)	1,6249	0,3692
Allergergische Rhinitis (SOLAR)	3,1184	0,2612
Geschlecht (*Referenz: männlich*)	-0,1001	0,2328
Sozioökonomischer Status (*Referenz: niedrig*)	-0,0870	0,2316
Als Säugling gestillt	-0,4534	0,2994
HMW-Exposition binär	-0,9426	0,4404
LMW-Exposition binär	0,6444	0,4083
MIXED-Exposition binär	0,7636	0,5970
IRRPEAKS-Exposition binär	-1,5730	1,1077
IRRPEAKS-Exposition binär	-0,5143	0,3050

Abbildung 7.5: Kombination der Parameterschätzer

Schätzer kombinieren - Modell für “Allergische Rhinitis in SOLAR II”

Tabelle 7.8 gibt eine Übersicht über die kombinierten Parameterschätzer und die zugehörigen kombinierten geschätzten Standardabweichungen für das Modell mit “Allergische Rhinitis in SOLAR II” als Zielgröße. [1]

[1] In den Tabellen 7.8 bis 7.11 werden einheitlich 5 Nachkommastellen dargestellt, um bei den Expositionsvariablen eine Tendenz erkennen zu können. (Dadurch sollen keine Aussagen über die Schätzgenauigkeit getroffen werden.)

Parameter	**Parameterschätzer**	**Standardabweichung**
Intercept	-2.40722	0.33747
Atopie der Eltern	0.59018	0.23244
Allerg.Rhinitis (ISAAC II)	1.62491	0.36921
Allerg.Rhinitis (SOLAR I)	3.11841	0.26118
Geschlecht *(Referenz: männlich)*	-0.10012	0.23283
Sozioökonom. Status *(Referenz: niedrig)*	-0.08702	0.23158
Als Säugling gestillt	-0.45340	0.29940
HMW-Exposition binär	-0.94256	0.44045
LMW-Exposition binär	0.64441	0.40832
MIXED-Exposition binär	0.76363	0.59700
IRRPEAKS-Exposition binär	-1.57299	1.10774
LOWRISK-Exposition binär	-0.51432	0.30500

Tabelle 7.8: Kombinierte Parameterschätzer und Standardabweichungen - Modell für Allergische Rhinitis in SOLAR II

Durch Exponieren der kombinierten Parameterschätzer wurden die Odds-Ratios für die kombinierten Parameterschätzer berechnet. Zusätzlich wurden noch 95%-Konfidenzintervalle für die Odds-Ratios berechnet.
In Tabelle 7.9 sind die Odds-Ratios sowie die Unter- und Obergrenzen der 95%-Konfidenzintervalle für die Odds-Ratios aufgeführt.

Parameter	**Untergrenze des 95%-KIs**	**Odds-Ratio**	**Obergrenze des 95%-KIs**
Intercept	0.04648	0.09007	0.17451
Atopie der Eltern	1.14406	1.80431	2.84558
Allerg.Rhinitis (ISAAC II)	2.46268	5.07796	10.47058
Allerg.Rhinitis (SOLAR I)	13.55150	22.61035	37.72480
Geschlecht *(Referenz: männlich)*	0.57323	0.90473	1.42793
Sozioökonom. Status *(Referenz: niedrig)*	0.58221	0.91666	1.44323
Als Säugling gestillt	0.35337	0.63546	1.14274
HMW-Exposition binär	0.16434	0.38963	0.92378
LMW-Exposition binär	0.85565	1.90487	4.24068
MIXED-Exposition binär	0.66598	2.14605	6.91538
IRRPEAKS-Exposition binär	0.02366	0.20742	1.81880
LOWRISK-Exposition binär	0.32886	0.59791	1.08706

Tabelle 7.9: Odds-Ratios und 95%-Konfidenzintervalle - Modell für Allergische Rhinitis in SOLAR II

In Abbildung 7.6 sind die Konfidenzintervalle der Odds-Ratios für die kombinierten Parameterschätzer für das Modell mit “Asthma in SOLAR II” als Zielgröße dargestellt.

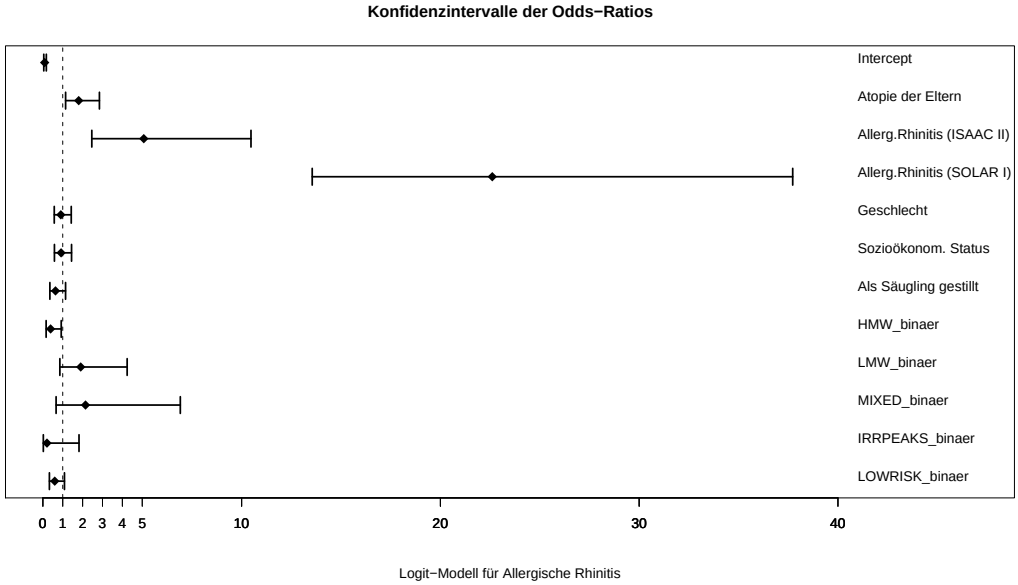


Abbildung 7.6: Konfidenzintervalle der Odds-Ratios - Modell für Allergische Rhinitis in SOLAR II

Schätzer kombinieren - Modell für "Asthma in SOLAR II"

Tabelle 7.10 gibt eine Übersicht über die kombinierten Parameterschätzer und die zugehörigen kombinierten geschätzten Standardabweichungen für das Modell mit "Asthma in SOLAR II" als Zielgröße.

Parameter	**Parameterschätzer**	**Standardabweichung**
Intercept	-4.07877	0.41546
Asthma (ISAAC II)	1.26855	0.51360
Geschlecht *(Referenz: männlich)*	-0.25624	0.35676
Neurodermitis (SOLAR I)	1.47098	0.36150
Allerg.Rhinitis (SOLAR I)	0.97875	0.36355
Asthma (SOLAR I)	2.78526	0.45968
Rauchen (SOLAR I)	0.72293	0.34387
Sozioökonom. Status *(Referenz: niedrig)*	-0.26866	0.34556
HMW-Exposition kumuliert	0.00026	0.00011
LMW-Exposition kumuliert	-0.00005	0.00013
MIXED-Exposition kumuliert	-0.00002	0.00016
IRRPEAKS-Exposition kumuliert	0.00076	0.00042
LOWRISK-Exposition kumuliert	-0.00001	0.00010

Tabelle 7.10: Kombinierte Parameterschätzer und Standardabweichungen - Modell für Asthma in SOLAR II

Durch Exponieren der kombinierten Parameterschätzer wurden die Odds-Ratios für die kombinierten Parameterschätzer berechnet. Zusätzlich wurden noch 95%-Konfidenzintervalle für die Odds-Ratios berechnet.
In Tabelle 7.11 sind die Odds-Ratios sowie die Unter- und Obergrenzen der 95%-Konfidenzintervalle für die Odds-Ratios aufgeführt.

Parameter	**Untergrenze des 95%-KIs**	**Odds-Ratio**	**Obergrenze des 95%-KIs**
Intercept	0.00750	0.01693	0.03822
Asthma (ISAAC II)	1.29938	3.55569	9.72993
Geschlecht *(Referenz: männlich)*	0.38463	0.77396	1.55738
Neurodermitis (SOLAR I)	2.14348	4.35350	8.84215
Allerg.Rhinitis (SOLAR I)	1.30499	2.66113	5.42656
Asthma (SOLAR I)	6.58164	16.20400	39.89425
Rauchen (SOLAR I)	1.05016	2.06046	4.04274
Sozioökonom. Status *(Referenz: niedrig)*	0.38831	0.76440	1.50476
HMW-Exposition kumuliert	1.00004	1.00026	1.00047
LMW-Exposition kumuliert	0.99970	0.99995	1.00019
MIXED-Exposition kumuliert	0.99968	0.99998	1.00029
IRRPEAKS-Exposition kumuliert	0.99994	1.00076	1.00158
LOWRISK-Exposition kumuliert	0.99978	0.99999	1.00019

Tabelle 7.11: Odds-Ratios und 95%-Konfidenzintervalle - Modell für Asthma in SOLAR II

In Abbildung 7.7 sind die Konfidenzintervalle der Odds-Ratios für die kombinierten Parameterschätzer für das Modell mit “Asthma in SOLAR II” als Zielgröße dargestellt.

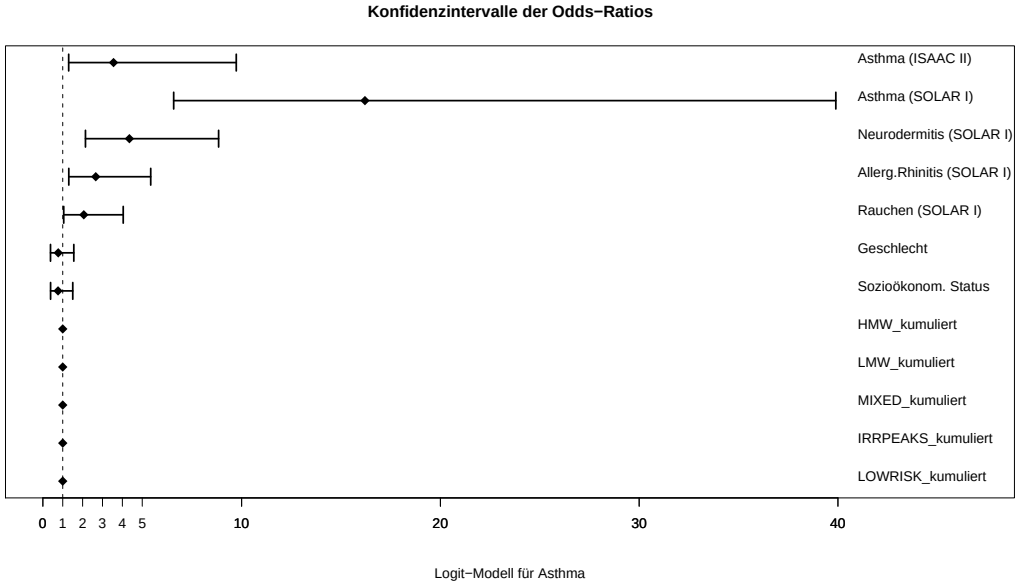


Abbildung 7.7: Konfidenzintervalle der Odds-Ratios - Modell für Asthma in SOLAR II

7.8.5 Interpretation der Odds-Ratios der kombinierten Parameterschätzer

Die Modellinterpretation erfolgt hier durch die Interpretation der Odds-Ratios der kombinierten Parameterschätzer (vgl. Beispiel zur Interpretation der Odds-Ratios in Abschnitt 7.3).

Interpretation - Modell für "Allergische Rhinitis in SOLAR II"

Die Chance in SOLAR II Allergische Rhinitis zu haben...

- ... ist für Personen, die mindestens ein atopisches Elternteil (d.h. Elternteil mit Neurodermitis, Allergische Rhinitis oder Asthma) haben, in etwa um den Faktor 1.8 erhöht im Gegensatz zur Chance von Personen, die kein atopisches Elternteil haben.
- ... ist für Personen, die in ISAAC II Allergische Rhinitis hatten, in etwa um den Faktor 5.1 erhöht im Gegensatz zur Chance von Personen, die in ISAAC II keine Allergische Rhinitis hatten.
- ... ist für Personen, die in SOLAR I Allergische Rhinitis hatten, in etwa um den Faktor 22.6 erhöht im Gegensatz zur Chance von Personen, die in SOLAR I keine Allergische Rhinitis hatten.

Diskussion - Modell für "Allergische Rhinitis in SOLAR II"

Aufgrund von Vorüberlegungen wurden die beiden Variablen "Geschlecht" und "sozioökonomischer Status" fest als Kovariablen in das logistische Modell aufgenommen. Im Bezug auf die Zielgröße "Allergische Rhinitis in SOLAR II" konnte kein signifikanter Unterschied zwischen den Geschlechtern und zwischen "höherem" und "niedrigerem" sozioökonomischem Status nachgewiesen werden.
Ob zwischen Probanden, die als Säugling gestillt wurden und Probanden, die als Säugling nicht gestillt wurden in Bezug auf das Auftreten von allergischer Rhinitis in SOLAR II tatsächlich ein Unterschied besteht, konnte nicht abschließend geklärt werden.
Da für das "beste" Modell fünf Modelle, die auf fünf verschiedenen Datensätzen gefittet wurden, kombiniert wurden, konnte es sein, dass eine Variable auf einem der Datensätze einen signifikanten Einfluss hatte und deswegen in das kombinierte Modell mit aufgenommen wurde, der Effekt dieser Variable allerdings auf den anderen Datensätzen und auch im kombinierten Modell nicht nachzuweisen war. Bei der Variable "Als Säugling gestillt" war dies der Fall. Da man jedoch sicherheitshalber lieber eine zusätzliche, vielleicht irrelevante, Variable in das "beste" Modell aufnehmen wollte, als eine möglicherweise relevante Variable nicht als Einflussgröße im Modell zu berücksichtigen, entschied man sich für die Aufnahme dieser Variable.

Interpretation - Modell für "Asthma in SOLAR II"

Die Chance in SOLAR II Asthma zu haben...

- ... ist für Personen, die in ISAAC II Asthma hatten in etwa um den Faktor 3.6 erhöht im Gegensatz zur Chance von Personen, die in ISAAC II kein Asthma hatten.
- ... ist für Personen, die in SOLAR I Neurodermitis hatten, in etwa um den Faktor 4.4 erhöht im Gegensatz zur Chance von Personen, die in SOLAR I keine Neurodermitis hatten.
- ... ist für Personen, die in SOLAR I Allergische Rhinitis hatten, in etwa um den Faktor 2.7 erhöht im Gegensatz zur Chance von Personen, die in SOLAR I keine Allergische Rhinitis hatten.
- ... ist für Personen, die in SOLAR I Asthma hatten, in etwa um den Faktor 16.2 erhöht im Gegensatz zur Chance von Personen, die in SOLAR I kein Asthma hatten.
- ... ist für Personen, die in SOLAR I Raucher waren, in etwa um den Faktor 2.1 erhöht im Gegensatz zur Chance von Personen, die in SOLAR I Nichtraucher waren.
- ... erhöht sich in etwa um den Faktor 1.00026, wenn sich für eine Person die HMW-Exposition um eine Stunde erhöht

Diskussion - Modell für "Asthma in SOLAR II"

Die Variablen "Geschlecht" und "sozioökonomischer Status" wurden aufgrund von substanzwissenschaftlichen Überlegungen fest als Kovariablen ins Modell aufgenommen. Es konnte kein signifikanter Unterschied in Bezug auf die Zielgröße "Asthma in SOLAR II" zwischen den Geschlechtern und zwischen "höherem" und "niedrigerem" sozioökonomischem Status nachgewiesen werden.
Da sich ein signifikanter Effekt der fünf Expositionsvariablen für die kumulierte Exposition über alle Jahre und Tätigkeiten zeigte, wurden aufgrund von biologischer Plausibiliät alle fünf Expositionsvariablen ins Modell aufgenommen. Durch genauere Untersuchung stellte sich heraus, dass vor allem die Expositionsvariablen "HMW-Exposition kumuliert" und "IRRPEAKS-Exposition kumuliert" einen signifikanten Einfluss auf die Zielgröße hatten.
Auch wenn ein Odds-Ratio in der Nähe von 1, wie es bei den Expositionsvariablen vorlag, auf den ersten Blick als sehr gering erscheinen mag, ist dies nicht der Fall, da die

Expositionsvariablen metrische Einflussgrößen sind. Deswegen kann ein Odds-Ratio, das nur minimal größer als 1 ist, einen starken Effekt haben, da der Wert des Odds-Ratios mit dem Wert der Expositionsvariable exponiert wird.

Beispiele:
1.) Eine Person, die ein Jahr lang einen Beruf mit IRRPEAKS-Exposition für 40 Stunden pro Woche ausübte, war also insgesamt 2040 Stunden in dieser Kategorie exponiert (2040 Stunden = 4,25 · 40 Wochenstunden · 12 Monate). Vergleicht man diese Person mit einer Person, die keiner IRRPEAKS-Exposition ausgesetzt war (bei sonst gleichen Werten der restlichen Kovariablen), so ist das Asthmarisiko der Person mit einer IRRPEAKS-Exposition von 2040 Stunden zum Zeitpunkt SOLAR II um den Faktor 5.1 ($1.0008^{2040} = 5.1$*) höher als bei der Person ohne Exposition in der Kategorie IRRPEAKS.*
2.) Betrachtet man nun eine Person, die ein Jahr lang einen Beruf mit HMW-Exposition für 40 Stunden pro Woche ausübte, war diese insgesamt 2040 Stunden in dieser Kategorie exponiert (2040 Stunden = 4,25 · 40 Wochenstunden · 12 Monate). Vergleicht man diese Person mit einer Person, die keiner HMW-Exposition ausgesetzt war (bei sonst gleichen Werten der restlichen Kovariablen), so ist das Asthmarisiko der Person mit einer HMW-Exposition von 2040 Stunden zum Zeitpunkt SOLAR II um den Faktor 1.8 ($1.0003^{2040} = 1.8$*) höher als bei der Person ohne Exposition in der Kategorie HMW.*

7.8.6 Diskussion der logistischen Regressionsmodelle

Medizinische Variablen als Einflussgrößen der logistischen Modelle

In die in dieser Arbeit gerechneten logistischen Regressionsmodelle gingen medizinische Variablen als Einflussgrößen ein, die den Zustand einer Person bezüglich einer Krankheit (erkrankt / nicht erkrankt) zu verschiedenen Beobachtungszeitpunkten beschreiben. So gingen in das Modell für die Zielgröße “Allergische Rhinitis in SOLAR II” die beiden Variablen “Allergische Rhinitis in ISAAC II” und “Allergische Rhinitis in SOLAR I” als Kovariablen ein. Ebenso gingen für die Zielgröße “Asthma in SOLAR II” die Variablen “Asthma in ISAAC II” und “Asthma in SOLAR I” als Einflussgrößen in das Modell ein. Diese Kovariablen stellten sich als “Haupteinflussgrößen” der logistischen Modelle heraus, d.h. durch die Aufnahme dieser Kovariablen in die logistischen Modelle konnte der Erkrankungsstatus bezüglich der Zielgrößen “Allergische Rhintis in SOLAR II” und “Asthma in SOLAR II” relativ gut erklärt werden.
Problematisch ist hierbei, dass die zeitliche Abfolge zwischen Exposition und Erkrankung nicht ausreichend berücksichtigt wurde.
Es kann zum Beispiel sein, dass die Exposition, der ein Proband bis zum Zeitpunkt

SOLAR I ausgesetzt war, sich schon auf den Wert der Variable "Allergische Rhinitis in SOLAR I" bzw. "Asthma in SOLAR I" ausgewirkt hat, d.h. der Proband ist zum Beispiel aufgrund seiner bisherigen Exposition zum Zeitpunkt SOLAR I an Asthma erkrankt.
Dadurch, dass die Exposition bis zum Zeitpunkt SOLAR I (in Kombination mit der Exposition bis zum Zeitpunkt SOLAR II) und die Variable "Allergische Rhinitis in SOLAR I" bzw. "Asthma in SOLAR I" beide als Einflussgröße in das logistische Modell eingingen, wurde die zeitliche Abfolge dieser beiden Variablen nicht berücksichtigt.
Um dieses Problem in den Griff zu bekommen, könnte man zum Beispiel beim Modell für die Zielgröße "Allergische Rhinitis in SOLAR II" alle Probanden ausschließen, die zu mindestens einem der Zeitpunkte ISAAC II und SOLAR I bereits an allergischer Rhinitis erkrankt waren. Somit würde man eine reine Inzidenzanalyse durchführen.

Expositionsvariablen als Einflussgrößen der logistischen Modelle

Die Expositionsvariablen wurden in das logistische Modell für die Zielgröße "Asthma in SOLAR II" als lineare Einflussgrößen aufgenommen.
Da in jeder der fünf Expositionskategorien (HMW,LMW,MIXED,IRRPEAKS,LOWRISK) sehr viele Probanden keiner Exposition ausgesetzt waren und nur bei verhältnismäßig wenigen Probanden eine Exposition vorlag, könnte man die Expositionen auch derart modellieren, dass die Probanden mit keiner Exposition anders behandelt werden als die Probanden, welche einer Exposition ausgesetzt waren.

Hierzu könnte zum Beispiel die Exposition als Einflussgröße wie folgt modelliert werden:
Für jede der fünf Kategorien wird ein Indikator (I) eingeführt, der angibt, ob eine Exposition in der entsprechenden Kategorie vorlag oder nicht:

$$I(exponiert) = \begin{cases} 1, & exponiert\ (in\ dieser\ Kategorie) \\ 0, & nicht\ exponiert\ (in\ dieser\ Kategorie) \end{cases}$$

Darauf basierend könnte dann für jede Kategorie die Exposition folgendermaßen modelliert werden:

$$\beta \cdot (1 - I) + \gamma \cdot I \cdot Expo$$

wobei "Expo" die Exposition in Stunden in der entsprechenden Kategorie angibt. Somit würde für die nichtexponierten Probanden (I=0) nur der konstante Term β in die Modellgleichung eingehen und für die exponierten Probanden (I=1) der Term $\gamma \cdot Expo$.

Es würde somit zwischen Probanden mit keiner Exposition und exponierten Probanden unterschieden werden, die Exposition würde jedoch nach wie vor als linearer Term in die Modellgleichung eingehen.

Alternativ könnte die Exposition auch nichtlinear modelliert werden, indem man die Exposition in bestimmte Kategorien einteilt, wie es bereits in den Studien ISAAC II [Riu et al. 2007] und SOLAR I durchgeführt wurde:

- Tätigkeiten mit hohem Asthmarisiko
  (Expositionskategorien HMW,LMW,MIXED,IRRPEAKS)
- Tätigkeiten mit niedrigem Asthmarisiko
  (Expositionskategorie LOWRISK)
- Tätigkeiten mit keiner asthmaspezifischen Exposition laut JEM
- Nie gearbeitet (und deswegen keiner Exposition ausgesetzt)

Die so entstandenen Kategorien könnten dann entweder dummykodiert oder als ordinale Einflussgrößen in die Modelle eingehen. Als Referenzkategorie bei Anwendung der Dummykodierung würden dann diejenigen Personen betrachtet werden, die keiner Exposition ausgesetzt waren.
Um dieses Vorgehen durchführen zu können, muss jedoch in jeder Kategorie eine ausreichend große Zahl von Probanden vorhanden sein. Es muss also auf Basis des entsprechenden Datensatzes überprüft werden, ob das Vorgehen realisierbar ist.

KAPITEL 8

Simulation

Es wurde nun eine Simulation durchgeführt, um Imputationsmethoden für Tätigkeitsdaten, die fehlende Werte enthalten, zu bewerten. Als Basis für die Simulation dienten die fünf Datensätze, die jeweils die imputierten Confoundervariablen und die Tätigkeitsdaten der 1094 Probanden mit vollständigen Tätigkeitsdaten enthalten. Abbildung 8.1 zeigt das Vorgehen bei der Simulation. Hier wird graphisch veranschaulicht, wie auf jedem der fünf Datensätze, welche jeweils die Daten der Probanden mit vollständigen Tätigkeitsangaben enthalten, vorgegangen wurde.

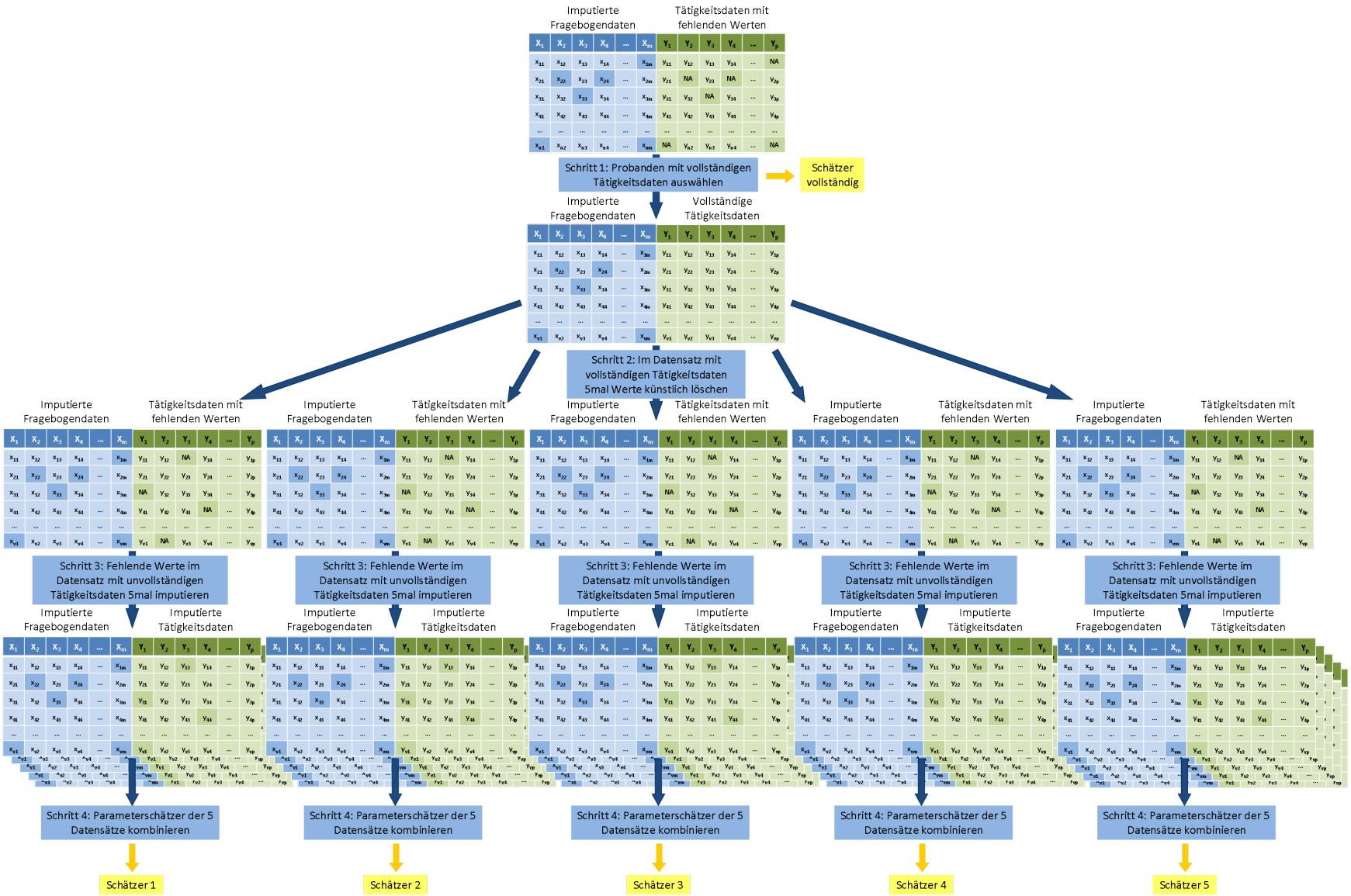


Abbildung 8.1: Vorgehen bei der Simulation auf jedem der fünf Datensätze mit imputierten Confoundervariablen und vollständigen Tätigkeitsdaten

8.1 Erzeugen eines Fehlendmusters in den Tätigkeitsdaten

Im Rahmen der Tätigkeitsdaten wurde neben der Tätigkeit und Branche, in der gearbeitet wurde, auch der Beginn der Tätigkeit (Anfangsmonat, Anfangsjahr), das Ende der Tätigkeit (Endmonat, Endjahr) und die Anzahl der Wochenstunden, die in der entsprechenden Tätigkeit gearbeitet wurden, abgefragt.

Unter Berücksichtigung des Schemas der fehlenden Werte in den Tätigkeitsangaben im Datensatz, der alle 1187 Probanden (d.h. 11870 Zeilen) enthält, wurde nun in den fünf Datensätzen, welche jeweils die Daten der 1094 Probanden mit vollständigen Tätigkeitsangaben enthalten, je fünf mal zufällig ein entsprechendes Fehlendmuster erzeugt. Tabelle 8.1 zeigt das Schema der fehlenden Daten im Datensatz, der alle 1187 Probanden enthält.

Fehlendmuster (11870 Zeilen)	**Anzahl Zeilen**
alle Angaben bis auf ISCO-Code fehlen	13 Zeilen (0,110 %)
nur Wochenstunden fehlen	27 Zeilen (0,227 %)
Wochenstunden, Zeitangaben zum Ende der Tätigkeit fehlen	14 Zeilen (0,118 %)
Zeitangaben zum Anfang und Ende der Tätigkeit fehlen	14 Zeilen (0,118 %)
Anfangsmonat und Endmonat fehlen	17 Zeilen (0,144 %)
Zeitangaben bis auf Anfangsjahr fehlen	15 Zeilen (0,126 %)
Anfangsjahr und Endjahr fehlen	1 Zeile (0,008 %)
nur Anfangsmonat fehlt	1 Zeile (0,008 %)
Wochenstunden und Zeitangaben bis auf Anfangsjahr fehlen	3 Zeilen (0,025 %)
Anfangsmonat und Wochenstunden fehlen	1 Zeile (0,008 %)
Zeitangaben bis auf Anfangsmonat fehlen	1 Zeile (0,008 %)
nur Endjahr fehlt	1 Zeile (0,008 %)
$\sum$	108 Zeilen

Tabelle 8.1: Fehlendmuster (Datensatz mit Tätigkeitsangaben aller Probanden)

Die Anzahlen der Zeilen, in denen eine bestimmte Kombination von Werten aus den Tätigkeitsangaben fehlt, wurden nun so umgerechnet, dass sich prozentual in etwa das gleiche Fehlendmuster auf jedem der Datensätze mit den 1094 Probanden (10940 Zeilen) mit vollständigen Tätigkeitsangaben ergab. Tabelle 8.2 gibt die Anzahl der Zeilen im Datensatz mit den 10940 Zeilen an, die gemäß der prozentualen Umrechnung eine bestimmte Kombination von fehlenden Werten aufweisen sollten.

Fehlendmuster (10940 Zeilen)	**Anzahl Zeilen**
alle Angaben bis auf ISCO-Code fehlen	12 Zeilen
nur Wochenstunden fehlen	25 Zeilen
Wochenstunden, Zeitangaben zum Ende der Tätigkeit fehlen	13 Zeilen
Zeitangaben zum Anfang und Ende der Tätigkeit fehlen	13 Zeilen
Anfangsmonat und Endmonat fehlen	16 Zeilen
Zeitangaben bis auf Anfangsjahr fehlen	14 Zeilen
Anfangsjahr und Endjahr fehlen	1 Zeile
nur Anfangsmonat fehlt	1 Zeile
Wochenstunden und Zeitangaben bis auf Anfangsjahr fehlen	3 Zeilen
Anfangsmonat und Wochenstunden fehlen	1 Zeile
Zeitangaben bis auf Anfangsmonat fehlen	1 Zeile
nur Endjahr fehlt	1 Zeile
$\sum$	101 Zeilen

Tabelle 8.2: Fehlendmuster (Datensatz der Probanden mit vollständigen Tätigkeitsangaben)

Nach diesem Fehlendmuster wurden nun in jedem der fünf Datensätze mit den Daten der 1094 Probanden (10940 Zeilen) mit vollständigen Tätigkeitsangaben fünf mal künstlich fehlende Werte erzeugt, d.h., bestimmte Werte wurden gelöscht. Um künstlich fehlende Werte zu erzeugen, wurden für jeden der fünf Datensätze fünf mal 101 Zeilen (Anzahl der Zeilen in denen ein oder mehrere Werte in den Tätigkeitsangaben fehlen) zufällig ohne Zurücklegen aus den 10940 Zeilen der Probanden mit vollständigen Tätigkeitsangaben nur aus denjenigen Zeilen gezogen, in denen Tätigkeitsangaben vorhanden sind. Dann wurden nach dem obigen Fehlendmuster künstlich die entsprechenden fehlende Werte erzeugt. Für jeden der fünf Datensätze mit den Daten der 1094 Probanden mit vollständigen Tätigkeitsangaben entstanden also fünf Datensätze, die künstlich gelöschte Werte enthalten. Es lagen nun also insgesamt 25 Datensätze mit künstlich gelöschten Werten in den Tätigkeitsangaben vor.

Da zufällig gezogen wurde und die Wahrscheinlichkeit für das Fehlen der Werte somit weder von der Variable, welche fehlende Werte enthält noch von irgendeiner anderen Variable abhängig war, ist der Fehlendmechanismus hier MCAR (“missing completely at random”).

8.2 Imputation der fehlenden Werte in den Tätigkeitsdaten

Nachdem nun künstlich fehlende Werte in den Tätigkeitsdaten erzeugt wurden, wurden die Tätigkeitsdaten durch bestimmte Imputationsmethoden, die im Abschnitt 8.2.1 beschrieben werden, wieder vervollständigt.
Für die Imputation wurde eine selbst geschriebene Funktion verwendet, der als Argumente ein Datensatz mit fehlenden Werten in den Tätigkeitsangaben und ein Startwert (zur Reproduzierbarkeit) übergeben wurden. Die Funktion lieferte einen vervollständigten Datensatz zurück.
Pro Datensatz mit fehlenden Werten in den Tätigkeitsangaben wurde fünf mal imputiert, um Variabilität zu erzeugen. Auf Basis eines Datensatzes mit unvollständigen Tätigkeitsangaben enstanden durch Imputation der fehlenden Werte in den Tätigkeitsangaben fünf Datensätze mit vervollständigten Tätigkeitsangaben. Es lagen anschließend 125 Datensätze vor, welche durch die Imputation vervollständigte Tätigkeitsdaten enthielten. Um festzustellen, wie gut die Imputationsmethoden funktionieren, wird später das logistische Modell, das auf den vollständigen Tätigkeitsdaten gerechnet wurde, erneut auf den imputierten Datensätzen berechnet und die Parameterschätzer werden geeignet kombiniert.

8.2.1 Vorgehen bei der Imputation

Als einfachste Möglichkeit bei der Imputation der fehlenden Tätigkeitsdaten könnte die Methode Ziehen aus der empirischen Verteilung angewandt werden. Zum Beispiel könnten dann die Wochenstunden direkt aus der empirischen Verteilung aller vorhandenen Wochenstundenangaben gezogen werden. Da dabei aber mögliche Einflussgrößen für den zu imputierenden Wert nicht berücksichtigt würden, jedoch zu vermuten ist, dass bestimmte Parameter einen Einfluss auf die Tätigkeitsdaten haben könnten, wurde diese Möglichkeit zunächst verworfen.
Es wurde untersucht, ob sich die Zeitangaben mit Hilfe eines linearen Modells mit den Prädiktoren “Alter”, “Geschlecht”, “sozioökonomischer Status” und “Berufssituation” (d.h. Schüler, Angestellt, Selbstständig, Arbeitslos etc.) vorhersagen lassen. Es wurden drei Modelle gerechnet, die jeweils die Einflussgrößen Geschlecht und sozioökonomischer Status hatten. Im ersten Modell (M0) gingen als zusätzliche Einflussgrößen die Berufssituation in SOLAR I, die Berufssituation in SOLAR II und das Alter zum Zeitpunkt von SOLAR II ein. Im zweiten Modell (M1) gingen als zusätzliche Einflussgrößen die Berufssituation in SOLAR I und das Alter zum Zeitpunkt in SOLAR I ein. Im dritten Modell (M2) gingen die Berufssituation aus SOLAR II und das Alter zum Zeitpunkt von SOLAR II als zusätzliche Einflussgrößen in das Modell ein.
Die Einflussgrößen der Modelle werden in Tabelle 8.3 nochmal als Übersicht dargestellt.

Dabei waren die mit einem "+" gekennzeichneten Einflussgrößen im Modell enthalten, die mit einem "-" gekennzeichneten Einflussgrößen nicht.

Einflussgröße	**Modell M0**	**Modell M1**	**Modell M2**
Geschlecht	+	+	+
Sozioökon. Status	+	+	+
Berufssituation in SOLAR I	+	+	-
Berufssituation in SOLAR II	+	-	+
Alter (zum Zeitpunkt von SOLAR I)	-	+	-
Alter (zum Zeitpunkt von SOLAR II)	+	-	+

Tabelle 8.3: Einflussgrößen der Modelle

Da das Bestimmtheitsmaß (R^2) bei allen drei Modellen sehr geringe Werte aufwies (R^2 im Bereich von 0.02 bis 0.10), wurde entschieden, die fehlenden Tätigkeitsangaben nicht mit Hilfe eines linearen Modells vorherzusagen.
Außerdem wurde festgelegt, dass die Imputation der Tätigkeitsdaten für SOLAR I und SOLAR II getrennt voneinander durchgeführt werden soll. Somit galten als Basis für die Imputation in der jeweiligen Studie ausschließlich die Angaben, die auch im Rahmen der entsprechenden Studie erhoben wurden.
Zudem entschied man, dass die fehlenden Zeitangaben und Wochenstunden aus der empirischen Verteilung gezogen werden, dabei allerdings zusätzlich nach bestimmten Variablen geschichtet wird. Nach welchen Variablen bei der Ziehung jeweils geschichtet wurde, wird im Folgenden ausführlich dargestellt.

8.2.2 Imputation der Zeitangaben

Bei der Imputation der Zeitangaben (Anfangsmonat, Anfangsjahr, Endmonat und Endjahr) wurde jeweils aus der empirischen Verteilung bedingt auf bestimmte Variablen gezogen. Dafür wurden aus der Liste der Kovariablen der linearen Modellen diejenigen Variablen ausgewählt, die zumindest bei einem der drei Modelle zu einem Niveau von 0.05 statistisch signifikant waren. Bei Faktorvariablen musste mindestens eine Faktorstufe zu diesem Niveau signifikant sein, um in die Auswahl aufgenommen zu werden. Tabelle 8.4 gibt eine Übersicht, welche Einflussgrößen bei welchen Zeitangaben (Anfangsmonat, Anfangsjahr, Endmonat und Endjahr) diese Vorgabe erfüllten.

Einflussgröße	Anfangsmonat	Anfangsjahr	Endmonat	Endjahr
Geschlecht	-	-	-	-
Sozioökon. Status	-	+	-	+
Berufssituation (SOLAR I)	-	-	-	-
Berufssituation (SOLAR II)	+	+	+	+
Alter (SOLAR I)	-	-	-	-
Alter (SOLAR II)	-	-	-	-

Tabelle 8.4: Einflussgrößen der Modelle bzgl. Zeitangabenimputation

Auf Basis dieser Ergebnisse wurde entschieden, das Ziehen aus der empirischen Verteilung (innerhalb der jeweiligen Studie) für alle Zeitangaben einheitlich durchzuführen. In SOLAR I wurde auf den sozioökonomischen Status bedingt, d.h., die Imputation erfolgte durch Ziehen aus der empirischen Verteilung geschichtet nach dem soziökonomischen Status. Da die Berufssituation aus SOLAR II zum Zeitpunkt von SOLAR I noch nicht bekannt war, wurde diese Variable für die Imputation der fehlenden Werte in SOLAR I nicht verwendet.

Für die Imputation der Zeitangaben in SOLAR II wurde aus der empirischen Verteilung geschichtet nach sozioökonomischem Status und der Berufssituation in SOLAR II gezogen.

Bei der Imputation der Zeitangaben musste zudem darauf geachtet werden, dass die imputierten Werte nicht zu unplausiblen Zeitangaben führen. So musste das Endjahr immer gleich dem Anfangsjahr oder in einem späteren Jahr sein. War bei der Imputation des Endjahrs der Anfangsmonat später als der Endmonat, so musste das Endjahr in einem späteren Jahr als das Anfangsjahr sein, es durfte nicht vor dem Anfangsjahr liegen und auch nicht gleich dem Anfangsjahr sein. Waren Anfangsjahr und Endjahr gleich, so musste der Endmonat gleich dem Anfangsmonat oder in einem späteren Monat sein. Diese Bedingungen wurden durch Anwendung der Methode “rejection sampling” verwirklicht, d.h., es wurde jeweils so lange gezogen bis ein Wert gezogen wurde, der die jeweilige Bedingung erfüllte.

Konnte bei der Imputation des Endjahrs in SOLAR II bei Bedingen auf den sozioökonomischen Status und der Berufssituation kein Endjahr gezogen werden, das gleich dem Anfangsjahr oder später als das Anfangsjahr war, so wurde nur auf den sozioökonomischen Status bedingt. Beim Endmonat wurde analog vorgegangen.

Die Abbildung 8.2 soll das Vorgehen am Beispiel der Imputation des Anfangsjahrs veranschaulichen. Hier wird für einen der Datensätze, der künstlich gelöschte Werte in den Tätigkeitsangaben enthält, für SOLAR I das Ziehen der Anfangsjahre bedingt auf den sozioökonomischen Status dargestellt.

Die Histogramme zeigen die empirische Verteilung des Anfangsjahrs in SOLAR I ge-

schichtet nach dem sozioökonomischen Status. Wie der Abbildung zu entnehmen ist wurde dabei nur aus den Anfangsjahren gezogen, die auch wirklich im Datensatz realisiert waren und es wurde jeweils mit den Häufigkeiten der Anfangsjahre gezogen, die aus den Daten ermittelt wurden.

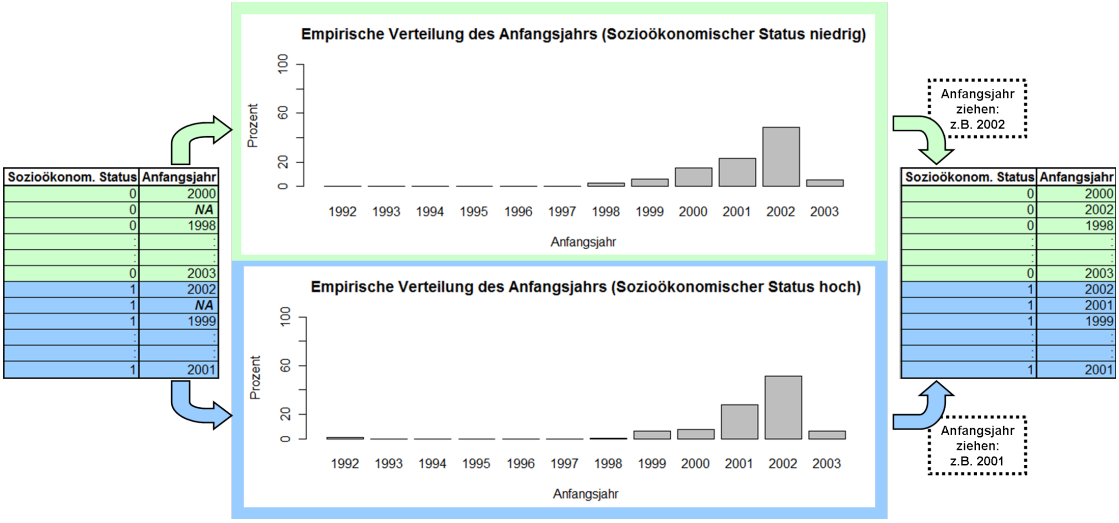


Abbildung 8.2: Imputation des Anfangsjahrs durch Ziehen aus der empirischen Verteilung geschichtet nach dem sozioökonomischen Status

8.2.3 Imputation der Wochenstunden

Die Imputation der Wochenstunden erfolgte durch Ziehen aus der empirischen Verteilung bedingt auf den ISCO-Code und auf das Geschlecht. Auf den ISCO-Code wurde bedingt, da die Anzahl der Wochenstunden in den verschiedenen Tätigkeitsgruppen sehr unterschiedlich sein kann. Die durchschnittliche Anzahl der Wochenstunden im Gastronomiegewerbe unterscheidet sich zum Beispiel oftmals deutlich von den Wochenstunden in anderen Tätigkeitsgruppen.
Um auch einen möglichen Geschlechtseffekt zu berücksichtigen, wurde zusätzlich noch auf das Geschlecht bedingt. Ein möglicher geschlechtsspezifischer Unterschied bei der Wochenstundenanzahl ist beim Vergleich der Mittelwerte der Wochenstundenanzahl (auf Basis der vollständigen Angaben) zu vermuten: Frauen bzw. Mädchen arbeiteten durchschnittlich 26, Männer bzw. Jungen 30 Stunden pro Woche.
Gab es bei Bedingen auf den ISCO-Code und das Geschlecht keine anderen Zeilen, in denen der gleiche ISCO-Code und das gleiche Geschlecht vorlag wie im Fall mit fehlenden Wochenstunden, so wurde nur auf das Geschlecht bedingt.

8.3 Logistische Regressionsmodelle auf imputierten Tätigkeitsdaten

Auf den 125 Datensätzen, welche die imputierten Tätigkeitsdaten enthalten, wurden nun noch einmal die logistischen Regressionsmodelle gerechnet, die in Kapitel 7 als "beste Modelle" für die Zielgrößen "Asthma in SOLAR II" und "Allergische Rhinitis in SOLAR II" ausgewählt wurden. Dafür mussten zuerst auf Basis der vervollständigten Datensätze die Expositionsvariablen, wie in Kapitel 6 beschrieben wurde, berechnet werden, die in die beiden Modelle als Einflussgrößen eingehen. Für das logistische Regressionsmodell mit "Allergische Rhinitis in SOLAR II" als Zielgröße wurden wiederum diejenigen Probanden ausgeschlossen, die in ISAAC II oder SOLAR I Asthma hatten.

8.4 Vergleich der Parameterschätzer

Nun sollten die Parameterschätzer der Modelle, die auf den Daten der Probanden mit vollständigen Tätigkeitsangaben angepasst wurden (in Abb. 8.1 als "Schätzer vollständig" dargestellt) mit den kombinierten Parameterschätzern verglichen werden, die durch die Berechnung der gleichen logistischen Modelle auf den Daten, welche die imputierten Tätigkeitsdaten enthalten, und anschließendes Kombinieren resultierten (in Abb. 8.1 als "Schätzer 1" bis "Schätzer 5" dargestellt).
Es wurden hierbei jeweils nur die Schätzer kombiniert, die zu einem Datensatz in dem Werte künstlich gelöscht wurden "gehörten". Das heisst pro Datensatz mit gelöschten Werten wurden fünf Schätzer kombiniert. Diese wurden anschließend mit dem Schätzer verglichen, der auf Basis der Daten mit den vollständigen Tätigkeitsangaben berechnet wurde. Der Vergleich der Parameterschätzer erfolgte auf Basis der Odds-Ratios der Parameterschätzer und wurde beispielhaft für das Modell für die Zielgröße Asthma in SOLAR II durchgeführt. Beim Modell für "Allergische Rhinitis in SOLAR II" ergeben sich ähnliche Bilder. Die folgenden Abbildungen zum Vergleich der Schätzer beziehen sich also alle auf das Modell für die Zielgröße "Asthma in SOLAR II".

In den Abbildungen für den Vergleich der Parameterschätzer auf Basis der Odds-Ratios der Schätzer ist das Odds-Ratio des Schätzers, der auf den imputierten Confoundervariablen und vollständigen Tätigkeitsdaten berechnet wurde, jeweils als roter Stern abgebildet. Dieses Odds-Ratio wird für den Vergleich der Odds-Ratios der Parameterschätzer als Odds-Ratios des "wahren Schätzers" angenommen, da man nur die Imputationsmethoden für die Tätigkeitsdaten betrachten und qualitativ bewerten möchte. Die Odds-Ratios der Parameterschätzer der Modelle, die auf den imputierten Tätigkeitsdaten berechnet wurden, sind als schwarze Kreise dargestellt. Die Skala der y-Achse wurden jeweils an den Bereich, in dem die Odds-Ratios liegen, angepasst. Eine Auswahl der Abbildungen für den Vergleich der Odds-Ratios der Schätzer wird in diesem Abschnitt behandelt.

Die restlichen Abbildungen sind im Anhang enthalten.

Vergleicht man pro Kovariable des logistischen Regressionsmodells das Odds-Ratio des "wahren" Parameterschätzers mit den Odds-Ratios der Schätzer, die auf Basis der imputierten Tätigkeitsdaten berechnet wurden und geht davon aus, dass die Imputationsmethoden für die Tätigkeitsdaten gut geeignet sind, so würde man (rein intuitiv) davon ausgehen, dass die Odds-Ratios der Schätzer auf Basis der imputierten Tätigkeitsdaten wie in Abbildung 8.3 nur einen kleinen Abstand zu dem Odds-Ratio des Schätzers auf Basis der vollständigen Tätigkeitsdaten haben und ein Teil der Odds-Ratios über dem Wert des "wahren" Parameterschätzers und ein Teil der Odds-Ratios unter diesem Wert liegt. Das würde bedeuten, dass die Schätzung des Parameters durch die Imputation der Tätigkeitsangaben nicht in eine bestimmte Richtung verzerrt wird, sondern dass der Schätzer, der auf den imputierten Datensätzen berechnet wurde, durch die Imputation der Tätigkeitsdaten nur zufällig um den "wahren" Schätzer schwankt.

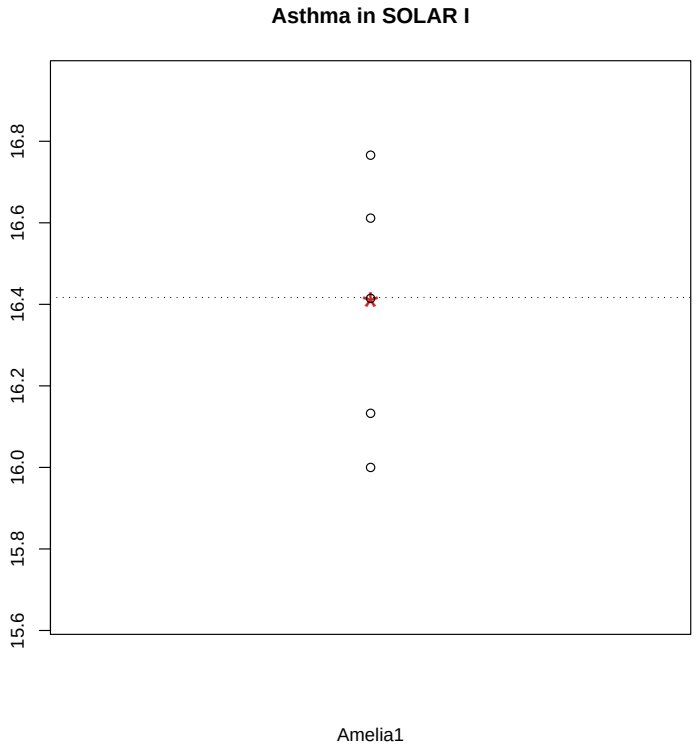


Abbildung 8.3: Vergleich der Parameterschätzer - Asthma in SOLAR I

Beunruhigend ist jedoch, dass zum Beispiel beim Vergleich der Odds-Ratios der Parameterschätzer für die Variable "IRRPEAKS-Exposition kumuliert" bei drei von fünf Datensätzen die Odds-Ratios der Schätzer, die auf den imputierten Tätigkeitsdaten berechnet wurden, ausschließlich über dem Wert des Odds-Ratios des "wahren" Schätzers liegen bzw. in etwa gleich dem Odds-Ratio des "wahren" Schätzer sind, und keine davon unter dem Odds-Ratio des "wahren" Wert des Schätzers liegt (vgl. erste drei Graphiken in Abb. 8.4).
Da der Wert der Expositionsvariable "IRRPEAKS-Exposition kumuliert" jedoch nur sehr selten (7 mal) größer als 0 war und die Schätzung des Parameters daher auf nur 7 Fällen mit einer Exposition in dieser Kategorie beruht, kann dieser Effekt eventuell durch die geringe Anzahl an Fällen erklärt werden.
Es ergibt sich jedoch auch beim Vergleich der Odds-Ratios der Parameterschätzer anderer Variablen ein ähnliches Bild, wie für die Einflussgröße Allergische Rhinitis in SOLAR I in der vierten Graphik in Abbildung 8.4 dargestellt ist. Hier liegen die Odds-Ratios der Schätzer, die auf den imputierten Datensätzen berechnet wurden, ausschließlich unter dem Wert des Odds-Ratios des "wahren" Schätzers. Allerdings ist das bei den Variablen außer "IRRPEAKS-Exposition kumuliert" nur jeweils auf 1-2 der 5 Datensätze mit vollständigen Berufsdaten und imputierten Confoundervariablen der Fall, dass sich ein solches Bild ergibt.

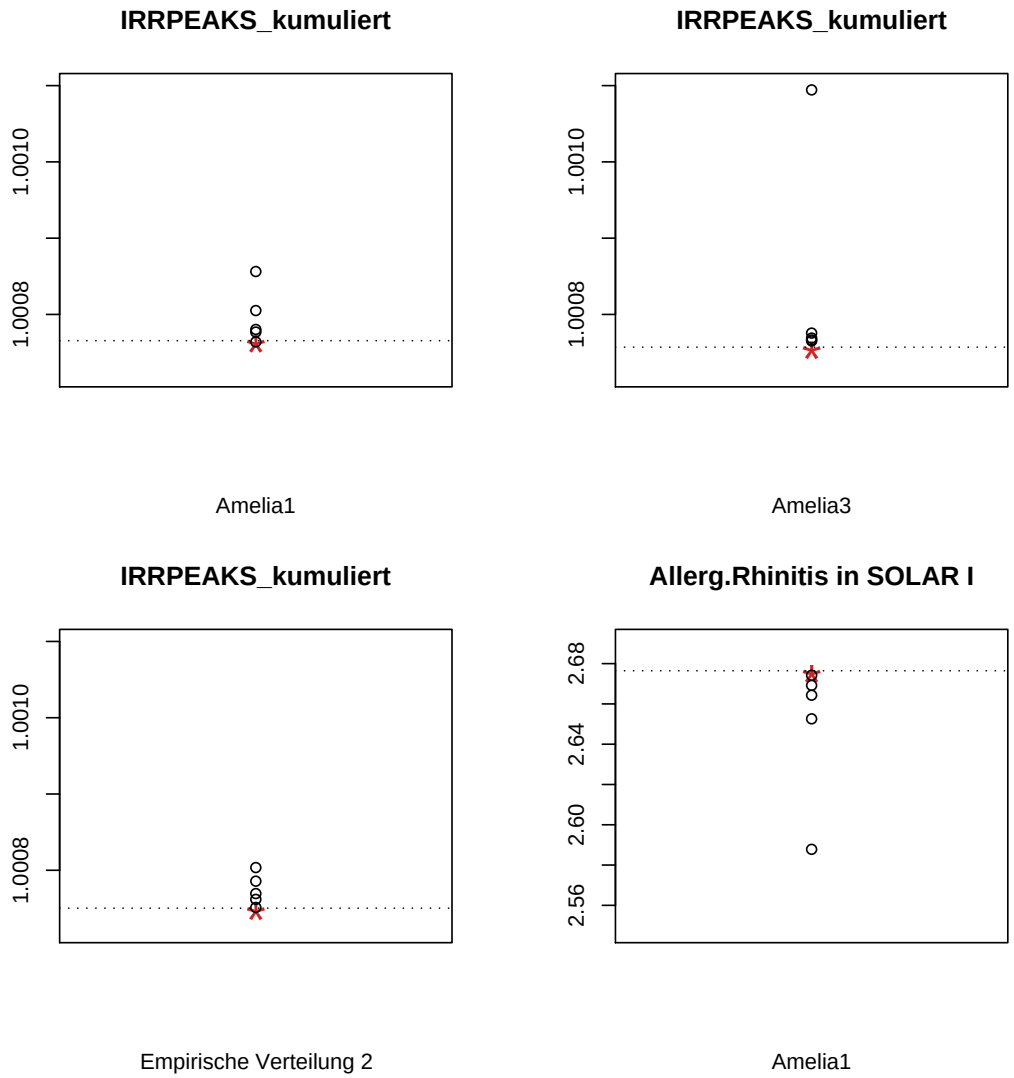


Abbildung 8.4: Vergleich der Parameterschätzer - "IRRPEAKS-Exposition kumuliert" bzw. Allergische Rhinitis in SOLAR I

Bei dem Odds-Ratio des Schätzers für die Expositionsvariable "LOWRISK-Exposition kumuliert" ist zum Beispiel für einen der Datensätze, die mit AMELIA II imputiert wurden, der Wert des Odds-Ratios des "wahren" Schätzers kleiner als 1. Der Wert von zwei der auf fünf den imputierten Tätigkeitsdaten berechneten Odds-Ratios ebenso. Die Werte der restlichen drei Odds-Ratios der Schätzer, die auf Basis der imputierten Tätigkeitsdaten berechnet wurden, sind größer als 1 (vgl. Abb. 8.5). Die Richtung des Effekts der Variable "LOWRISK-Exposition kumuliert" ist hier nicht eindeutig.

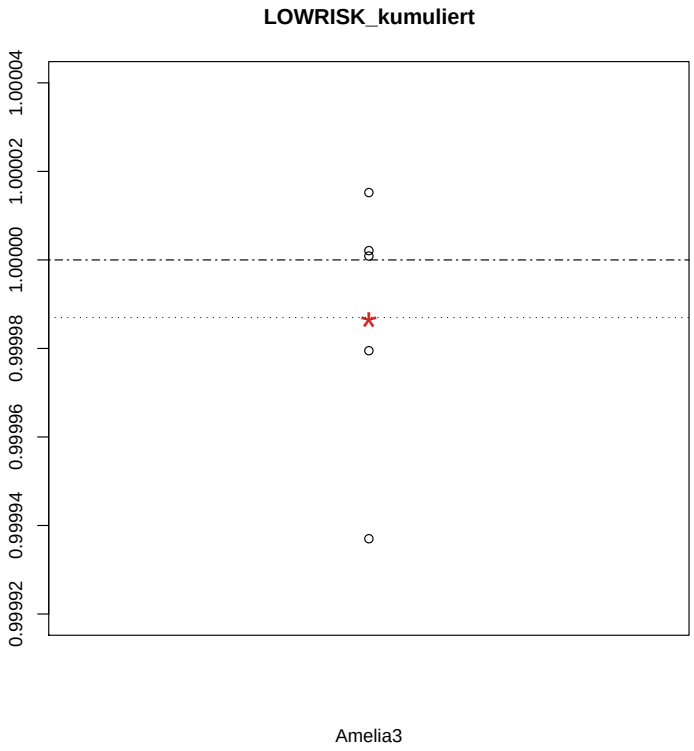


Abbildung 8.5: Vergleich der Parameterschätzer - LOWRISK_kumliert

Fazit

Im Allgemeinen ergibt sich jedoch beim Vergleich der Odds-Ratios der Parameterschätzer meist ein gleichmäßiges Bild, das in etwa dem in Abbildung 8.3 dargestellten entspricht. Obwohl es vereinzelt Fälle gibt, in denen die Odds-Ratios der Parameterschätzer, welche auf den imputierten Tätigkeitsdaten berechnet wurden, ausschließlich über bzw. unter dem Wert des Odds-Ratios des "wahren" Parameters liegen und im Fall der Variable "LOWRISK-Exposition kumuliert" der Effekt der Variable nicht eindeutig ist, unterliegen die Schätzer welche auf den imputierten Tätigkeitsdaten berechnet wurden meist nur Zufallsschwankungen, die durch die Imputation der Tätigkeitsdaten entstanden sind.

Berechnung der maximalen Abweichungen der Schätzer

Die maximalen Abweichungen der Odds-Ratios der Schätzer auf Basis der imputierten Tätigkeitsdaten vom Odds-Ratio des "wahren" Schätzers sind in den Tabellen 8.5 und 8.6 beispielhaft für das Modell für die Zielgröße "Asthma in SOLAR II" aufgeführt. [2]

Die maximale Abweichung wurde für jede Kovariable pro betrachtetem Datensatz berechnet. Bei den Confounder-Variablen ist die maximale Abweichung stets kleiner als 0.6 und bei den Expositionsvariablen kleiner als 0.0004.

(Beim Modell für die Zielgröße "Allergische Rhinitis in SOLAR II" ist die maximale Abweichung bei den Confounder-Variablen stets kleiner als 0.3 und bei den Expositionsvariablen kleiner als 0.4.)

Da diese maximale Abweichung bei allen Kovariablen relativ gering ist, kann man davon ausgehen, dass nur geringe Zufallsschwankungen der Schätzer auf den imputierten Tätigkeitsdaten um den "wahren" Schätzer durch Imputation der Tätigkeitsdaten entstanden sind.

[2] In den beiden Tabellen werden einheitlich 5 Nachkommastellen dargestellt, um bei den Expositionsvariablen eine Tendenz erkennen zu können. (Dadurch sollen keine Aussagen über die Schätzgenauigkeit getroffen werden.)

Parameter	Datensatz	Odds Ratio des "wahren" Schätzers	Maximale Abweichung
Intercept	1 (Amelia)	0.01775	0.00064
	2 (Amelia)	0.01712	0.00101
	3 (Amelia)	0.01694	0.00059
	4 (Empir.Vert.)	0.01658	0.00080
	5 (Empir.Vert.)	0.01629	0.00114
Asthma (ISAAC II)	1 (Amelia)	3.48646	0.31607
	2 (Amelia)	3.54363	0.16329
	3 (Amelia)	3.57649	0.08059
	4 (Empir.Vert.)	3.55435	0.07745
	5 (Empir.Vert.)	3.61882	0.06582
Geschlecht	1 (Amelia)	0.77265	0.01716
	2 (Amelia)	0.77375	0.00725
	3 (Amelia)	0.77269	0.01204
	4 (Empir.Vert.)	0.77687	0.01994
	5 (Empir.Vert.)	0.77385	0.01507
Neurodermitis (SOLAR I)	1 (Amelia)	4.35410	0.04248
	2 (Amelia)	4.35332	0.05923
	3 (Amelia)	4.36513	0.06006
	4 (Empir.Vert.)	4.33088	0.05380
	5 (Empir.Vert.)	4.36417	0.09224
Allerg.Rhinitis (SOLAR I)	1 (Amelia)	2.67646	0.08864
	2 (Amelia)	2.65870	0.02438
	3 (Amelia)	2.65451	0.02640
	4 (Empir.Vert.)	2.67135	0.02583
	5 (Empir.Vert.)	2.64474	0.07993
Asthma (SOLAR I)	1 (Amelia)	16.41678	0.41683
	2 (Amelia)	16.12662	0.45247
	3 (Amelia)	16.21339	0.51792
	4 (Empir.Vert.)	16.22476	0.36868
	5 (Empir.Vert.)	16.04085	0.13004
Rauchen (SOLAR I)	1 (Amelia)	2.04398	0.07323
	2 (Amelia)	2.05773	0.05965
	3 (Amelia)	2.06972	0.07436
	4 (Empir.Vert.)	2.06762	0.01894
	5 (Empir.Vert.)	2.06338	0.03092
Sozioökonom. Status	1 (Amelia)	0.70689	0.01936
	2 (Amelia)	0.75052	0.01968
	3 (Amelia)	0.76422	0.01023
	4 (Empir.Vert.)	0.78990	0.01467
	5 (Empir.Vert.)	0.81490	0.02263

Tabelle 8.5: Maximale Abweichung der Odds-Ratios der Schätzer auf den imputierten Tätigkeitsdaten vom Odds-Ratio des"wahren" Schätzers auf den vollständigen Tätigkeitsdaten - Confoundervariablen

Parameter	**Datensatz**	**Odds-Ratio des "wahren" Schätzers**	**Maximale Abweichung**
HMW-Exposition kumuliert	1 (Amelia)	1.00025	0.00003
	2 (Amelia)	1.00026	0.00002
	3 (Amelia)	1.00026	0.00005
	4 (Empir.Vert.)	1.00026	0.00003
	5 (Empir.Vert.)	1.00026	0.00002
LMW-Exposition kumuliert	1 (Amelia)	0.99995	0.00003
	2 (Amelia)	0.99995	0.00002
	3 (Amelia)	0.99995	0.00008
	4 (Empir.Vert.)	0.99995	0.00002
	5 (Empir.Vert.)	0.99995	0.00003
MIXED-Exposition kumuliert	1 (Amelia)	0.99998	0.00004
	2 (Amelia)	0.99998	0.00001
	3 (Amelia)	0.99998	0.00001
	4 (Empir.Vert.)	0.99999	0.00000
	5 (Empir.Vert.)	0.99999	0.00002
IRRPEAKS-Exposition kumuliert	1 (Amelia)	1.00077	0.00009
	2 (Amelia)	1.00076	0.00007
	3 (Amelia)	1.00076	0.00034
	4 (Empir.Vert.)	1.00075	0.00002
	5 (Empir.Vert.)	1.00075	0.00005
LOWRISK-Exposition kumuliert	1 (Amelia)	0.99999	0.00004
	2 (Amelia)	0.99999	0.00003
	3 (Amelia)	0.99999	0.00005
	4 (Empir.Vert.)	0.99999	0.00005
	5 (Empir.Vert.)	0.99999	0.00007

Tabelle 8.6: Maximale Abweichung der Odds-Ratios der Schätzer auf den imputierten Tätigkeitsdaten vom Odds-Ratio des"wahren" Schätzers auf den vollständigen Tätigkeitsdaten - Expositionsvariablen

KAPITEL 9

Zusammenfassung und Ausblick

9.1 Zusammenfassung

In dieser Arbeit wurde zuerst die Imputation fehlender Werte in den potentiellen Confoundervariablen durchgeführt, um anschließend ein Modell basierend auf diesen imputierten Confoundervariablen und den vollständigen Tätigkeitsdaten anpassen zu können. Bei der Modellwahl wurde in zwei Schritten vorgegangen: Zuerst wurde ein geeignetes Confoundermodell gewählt, das nur Variablen aus den potentiellen Confoundervariablen enthielt. Im zweiten Schritt wurde überprüft, ob zusätzlich Expositionsvariablen in das Modell aufgenommen werden sollen, ob die Aufnahme dieser Variablen also zu einer Modellverbesserung führt. Beim Modell für "Allergische Rhinitis in SOLAR II" führte die Aufnahme von Expositionsvariablen in das Confoundermodell zu keiner Modellverbesserung. Aus inhaltlichen Gründen wurden jedoch die Expositionsvariablen für die binäre Exposition über alle Tätigkeiten und Jahre in das Modell aufgenommen, wodurch man das "beste" Modell für die Zielgröße "Allergische Rhinitis in SOLAR II" erhielt. Die Einflussgrößen dieses "besten" Modells sind der folgenden Tabelle zu entnehmen.

Einflussgrößen des "besten" Modells für Allergische Rhinitis in SOLAR II:
Geschlecht, Sozioökonomischer Status, Atopie der Eltern,
Allergische Rhinits (ISAAC II),Allergische Rhinitis (SOLAR I),Als Säugling gestillt
HMW-Exposition binär, LMW-Exposition binär, MIXED-Exposition binär
IRRPEAKS-Exposition binär, LOWRISK-Exposition binär

Beim Modell für "Asthma in SOLAR II" führte die Aufnahme der Expositionsvariablen für die Exposition kumuliert über alle Tätigkeiten und Jahre zu einer Modellverbesserung. Die Exposition im ersten Tätigkeitsjahr und die Exposition in der ersten Tätigkeit

konnten hier nicht als Surrogat für die Exposition kumuliert über alle Tätigkeiten und Jahre verwendet werden. Durch die Aufnahme dieser Expositionsvariablen in das Modell erhielt man das "beste" Modell für die Zielgröße "Asthma in SOLAR II", dessen Einflussgrößen in der folgenden Tabelle aufgeführt sind.

Einflussgrößen des "besten" Modells für Asthma in SOLAR II:
Geschlecht, Sozioökonomischer Status, Rauchen (SOLAR I), Asthma (ISAAC II), Asthma (SOLAR I), Neurodermitis (SOLAR I), Allerg.Rhinitis (SOLAR I), HMW-Exposition kumuliert, LMW-Exposition kumuliert, MIXED-Exposition kumuliert IRRPEAKS-Exposition kumuliert, LOWRISK-Exposition kumuliert

Im Anschluss an die Modellwahl wurde eine Simulation durchgeführt, in der Imputationsmethoden für fehlende Werte in den Tätigkeitsdaten getestet werden sollten. Dazu wurden in den Tätigkeitsdaten künstlich fehlende Werte erzeugt, die anschließend durch bestimmte Methoden imputiert wurden. Hier konnte ein qualitativer Vergleich von Parameterschätzern durchgeführt werden.

9.2 Ausblick

Diese Arbeit hatte neben der Behandlung fehlender Daten und einer Simulation das Ziel, logistische Regressionsmodelle an die vorliegenden Daten aus der SOLAR-Kohortenstudie anzupassen und die Ergebnisse der Modelle zu interpretieren. Durch das AIC-Kriterium und die ROC-Analyse wurden Instrumente genutzt, um die Modellgüte auf den vorliegenden Daten zu beurteilen. Dabei wird jedoch bei der ROC-Analyse die AUC ("area under the curve") überschätzt. Legt man Wert darauf, nicht nur ein gutes Modell für die vorliegenden Daten sondern auch ein gutes Prognosemodell zu finden, so sollte zum Beispiel noch eine Kreuzvalidierung zur Modellevaluation durchgeführt werden. Bei einer Kreuzvalidierung werden die Daten aufgeteilt in einen Trainingsdatensatz, auf dem die Parameterschätzung durchgeführt wird, und einen Testdatensatz, mit dessen Hilfe die Prognosegüte abgeschätzt werden kann.

In dieser Arbeit wurde in Kapitel 8 eine Simulation durchgeführt, um Imputationsmethoden für Tätigkeitsdaten zu bewerten. Im Rahmen dieser Simulation wurden Vergleiche in Bezug auf Parameterschätzer durchgeführt. Aufgrund der limitierten zur Verfügung stehenden Zeit wurden pro Datensatz mit vollständigen Tätigkeitsdaten und imputierten Confoundervariablen fünfmal Werte aus den Tätigkeitsdaten künstlich gelöscht und anschließend wurde pro Datensatz mit künstlich gelöschten Werten fünfmal imputiert. So

konnte pro Datensatz ein “wahrer” Schätzer mit fünf kombinierten, auf Basis der imputierten Tätigkeiten berechneten Schätzern verglichen werden. Aufgrund dieser geringen Anzahl konnten die Parameterschätzer nur qualitativ verglichen werden. Als eine zukünftige Herausforderung, auch aus programmiertechnischer Sicht, könnte deshalb gesehen werden, in einer Simulation sehr viel öfter künstlich fehlende Werte in den Tätigkeitsangaben zu erzeugen und pro Datensatz mit künstlich gelöschten Werten sehr viel öfter zu imputieren, um anschließend einen quantitativen Vergleich der Parameterschätzer durchführen zu können.
In dieser Arbeit war der Fehlendmechanismus beim Erzeugen fehlender Werte im Datensatz MCAR. Interessant wäre auch, das Löschen mit dem Fehlendmechanismus MAR durchzuführen und die Ergebnisse zu vergleichen.
Eine weitere Idee wäre, auch die Imputationsmethoden für die Confoundervariablen anhand einer Simulation zu bewerten.

ANHANG A

Variablenkodierung

A.1 Variablen aus ISAAC II

A.1.1 In Deutschland geboren

Die Angaben zum Geburtsort aus ISAAC II wurden so kodiert, dass man die Unterscheidung treffen kann, ob jemand in Deutschland geboren ist oder nicht. “In Deutschland geboren” wird hier dadurch definiert, ob das Kind die deutsche Staatsangehörigkeit hat.

Verwendete Variablen:

Variablenname	**Frage & Kodierung**
STAATS1	Welche Staatsangehörigkeit hat Ihr Kind? 01=deutsch, 02=russisch, ..., 15=sonstiges, NA=Missing

Bildung der Variable d_geb

Kodierungsbeschreibung:
d_geb wird auf 1 gesetzt, wenn STAATS1 den Wert 1 hat.
d_geb wird auf missing (NA) gesetzt, wenn STAATS1 missing (NA) ist.
In allen anderen Fällen steht in der Variablen d_geb der Wert 0.

Kodierungsübersicht:

Verwendete Variablen	**Abgeleitete Variablen**
STAATS1	**geb_d**
1	1
2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15	0
NA	NA

Code-Übersicht:

Variablenname	**Frage & Kodierung**
geb_d	In Deutschland geboren 0=Nein, 1=Ja, NA=Missing

R-Code:

```
> d_geb <- 0
> d_geb[is.na(STAATS1)] <- NA
> d_geb[!is.na(STAATS1)&(STAATS1==1)] <- 1
```

A.1.2 Atopie der Eltern

Die Variable PAR_ALL aus ISAAC-II wird rekodiert, um eine 0-1-Kodierung zu erhalten, um die Interpretation und Vergleichbarkeit mit den (0-1-kodierten) Variablen aus SOLAR I und II zu vereinfachen.

Verwendete Variablen:

Variablenname	Frage & Kodierung
PAR_ALL	Atopie der Eltern Hatte mindestens ein Elternteil des Kindes irgendwann einmal Asthma, Heuschnupfen oder Neurodermitis? 1=Ja, 2=Nein, NA=Keine Angabe

Bildung der Variable PAR_ALL_r

Kodierungsbeschreibung:
PAR_ALL_r wird auf 0 (Nein) gesetzt, wenn PAR_ALL den Wert 2 hat.
PAR_ALL_r bleibt auf 1 (Ja) gesetzt, wenn PAR_ALL den Wert 1 hat.
PAR_ALL_r ist missing (NA), wenn PAR_ALL missing (NA) ist.

Kodierungsübersicht:

Verwendete Variablen	Abgeleitete Variablen
PAR_ALL	**PAR_ALL_r**
1	1
2	0
NA	NA

Code-Übersicht:

Variablenname	Frage & Kodierung
PAR_ALL_r	Atopie der Eltern 0=Nein, 1=Ja, NA=Missing

R-Code:

```
> PAR_ALL_r[PAR_ALL==2] <- 0
> PAR_ALL_r[PAR_ALL==1] <- 1
```

A.1.3 Kind gestillt

Die Variable STILL aus ISAAC II wird rekodiert, um eine 0-1-Kodierung zu erhalten, um die Interpretation und Vergleichbarkeit mit den (0-1-kodierten) Variablen aus SOLAR I und II zu vereinfachen.

Verwendete Variablen:

Variablenname	Frage & Kodierung
STILL	Kind gestillt (Wurde Ihr Kind gestillt?) 1=Ja, 2=Nein, NA=Keine Angabe

Bildung der Variable STILL_r

Kodierungsbeschreibung:
STILL_r wird auf 0 (Nein) gesetzt, wenn STILL den Wert 2 hat.
STILL_r bleibt auf 1 (Ja) gesetzt, wenn STILL den Wert 1 hat.
STILL_r ist missing (NA), wenn STILL missing (NA) ist.

Kodierungsübersicht:

Verwendete Variablen	Abgeleitete Variablen
STILL	**STILL_r**
1	1
2	0
NA	NA

Code-Übersicht:

Variablenname	Frage & Kodierung
STILL_r	Kind gestillt 0=Nein, 1=Ja, NA=Missing

R-Code:

```
> STILL_r[STILL==2] <- 0
> STILL_r[STILL==1] <- 1
```

A.1.4 Neurodermitis

Die Variable CUR_DERM aus ISAAC-II wird rekodiert, um eine 0-1-Kodierung zu erhalten, um die Interpretation und Vergleichbarkeit mit den (0-1-kodierten) Variablen aus SOLAR-I und -II zu vereinfachen.

Verwendete Variablen:

Variablenname	**Frage & Kodierung**
CUR_DERM	Neurodermitis 1=Ja, 2=Nein, NA=Keine Angabe

Bildung der Variable CUR_DERM_r

Kodierungsbeschreibung:
CUR_DERM_r wird auf 0 (Nein) gesetzt, wenn CUR_DERM den Wert 2 hat.
CUR_DERM_r bleibt auf 1 (Ja) gesetzt, wenn CUR_DERM den Wert 1 hat.
CUR_DERM_r ist missing (NA), wenn CUR_DERM missing (NA) ist.

Kodierungsübersicht:

Verwendete Variablen	**Abgeleitete Variablen**
CUR_DERM	**CUR_DERM_r**
1	1
2	0
NA	NA

Code-Übersicht:

Variablenname	**Frage & Kodierung**
CUR_DERM_r	Neurodermitis 0=Nein, 1=Ja, NA=Missing

R-Code:

```
> CUR_DERM_r[CUR_DERM==2] <- 0
> CUR_DERM_r[CUR_DERM==1] <- 1
```

A.1.5 Allergische Rhinitis

Die Variable CUR_HAY aus ISAAC II wird rekodiert, um eine 0-1-Kodierung zu erhalten, um die Interpretation und Vergleichbarkeit mit den (0-1-kodierten) Variablen aus SOLAR I und II zu vereinfachen.

Verwendete Variablen:

Variablenname	Frage & Kodierung
CUR_HAY	Allergische Rhinitis 1=Ja, 2=Nein, NA=Keine Angabe

Bildung der Variable CUR_HAY_r

Kodierungsbeschreibung:
CUR_HAY_r wird auf 0 (Nein) gesetzt, wenn CUR_HAY den Wert 2 hat.
CUR_HAY_r bleibt auf 1 (Ja) gesetzt, wenn CUR_HAY den Wert 1 hat.
CUR_HAY_r ist missing (NA), wenn CUR_HAY missing (NA) ist.

Kodierungsübersicht:

Verwendete Variablen	Abgeleitete Variablen
CUR_HAY	**CUR_HAY_r**
1	1
2	0
NA	NA

Code-Übersicht:

Variablenname	Frage & Kodierung
CUR_HAY_r	Allergische Rhinitis 0=Nein, 1=Ja, NA=Missing

R-Code:

```
> CUR_HAY_r[CUR_HAY==2] <- 0
> CUR_HAY_r[CUR_HAY==1] <- 1
```

A.1.6 Asthma

Die Variable CUR_ASTH aus ISAAC II wird rekodiert, um eine 0-1-Kodierung zu erhalten, um die Interpretation und Vergleichbarkeit mit den (0-1-kodierten) Variablen aus SOLAR I und II zu vereinfachen.

Verwendete Variablen:

Variablenname	**Frage & Kodierung**
CUR_ASTH	Asthma 1=Ja, 2=Nein, NA=Keine Angabe

Bildung der Variable CUR_ASTH_r

Kodierungsbeschreibung:
CUR_ASTH_r wird auf 0 (Nein) gesetzt, wenn CUR_ASTH den Wert 2 hat.
CUR_ASTH_r bleibt auf 1 (Ja) gesetzt, wenn CUR_ASTH den Wert 1 hat.
CUR_ASTH_r ist missing (NA), wenn CUR_ASTH missing (NA) ist.

Kodierungsübersicht:

Verwendete Variablen	**Abgeleitete Variablen**
CUR_ASTH	**CUR_ASTH_r**
1	1
2	0
NA	NA

Code-Übersicht:

Variablenname	**Frage & Kodierung**
CUR_ASTH_r	Asthma 0=Nein, 1=Ja, NA=Missing

R-Code:

```
> CUR_ASTH_r[CUR_ASTH==2] <- 0
> CUR_ASTH_r[CUR_ASTH==1] <- 1
```

A.1.7 Passivrauch

Die Variable ETSNOW aus ISAAC II wird rekodiert, so dass sie mit den Variablen aus SOLAR I und II einfacher vergleichbar ist.

Verwendete Variablen:

Variablenname	Frage & Kodierung
ETSNOW	Passivrauch Ist das Kind in der Wohnung Tabakrauch ausgesetzt? 1=Eltern zur Zeit Raucher, 2=Eltern ehemalige Raucher, 3=Eltern nie geraucht, NA=Keine Angabe

Bildung der Variable ETSNOW_r

Kodierungsbeschreibung:
ETSNOW_r bleibt auf 1 (Eltern zur Zeit Raucher) gesetzt, wenn ETSNOW den Wert 1 hat.
ETSNOW_r bleibt auf 2 (Eltern ehemalige Raucher) gesetzt, wenn ETSNOW den Wert 2 hat.
ETSNOW_r wird auf 0 (Eltern nie geraucht) gesetzt, wenn ETSNOW den Wert 3 hat.
ETSNOW_r ist missing (NA), wenn ETSNOW missing (NA) ist.

Kodierungsübersicht:

Verwendete Variablen	Abgeleitete Variablen
ETSNOW	**ETSNOW_r**
1	1
2	2
3	0
NA	NA

Code-Übersicht:

Variablenname	Frage & Kodierung
ETSNOW_r	Asthma 0=Eltern nie geraucht, 1=Eltern zur Zeit Raucher, 2=Eltern ehemalige Raucher, NA=Keine Angabe

R-Code:

```
> ETSNOW_r[ETSNOW==2] <- 2
> ETSNOW_r[ETSNOW==1] <- 1
> ETSNOW_r[ETSNOW==3] <- 0
```

A.1.8 Sozioökonomischer Status

Die Variable SES aus ISAAC II wird rekodiert, um eine 0-1-Kodierung zu erhalten, um die Interpretation und Vergleichbarkeit mit den (0-1-kodierten) Variablen aus SOLAR I und II zu vereinfachen.

Verwendete Variablen:

Variablenname	Frage & Kodierung
SES	Sozioökonomischer Status Schulabschluss (bzw. Dauer des Schulbesuchs) der Eltern 1=Hoch (Fachabitur/Abitur/Studium), 2=Niedrig (Niedrigere Ausbildung), NA=Keine Angabe

Bildung der Variable SES_r

Kodierungsbeschreibung:
SES_r wird auf 0 (Niedrig) gesetzt, wenn SES den Wert 2 hat.
SES_r bleibt auf 1 (Hoch) gesetzt, wenn SES den Wert 1 hat.
SES_r ist missing (NA), wenn SES missing (NA) ist.

Kodierungsübersicht:

Verwendete Variablen	Abgeleitete Variablen
SES	**SES_r**
1	1
2	0
NA	NA

Code-Übersicht:

Variablenname	Frage & Kodierung
SES_r	Sozioökonomischer Status 0=Niedirg (Niedrigere Ausbildung), 1=Hoch (Fachabitur/Abitur/Studium), NA=Missing

R-Code:

```
> SES_r[SES==2] <- 0
> SES_r[SES==1] <- 1
```

A.1.9 Studienzentrum

Die Variable zentrum aus ISAAC II wird rekodiert, um eine 0-1-Kodierung zu erhalten, um die Interpretation und Vergleichbarkeit mit den (0-1-kodierten) Variablen aus SOLAR I und II zu vereinfachen.

Verwendete Variablen:

Variablenname	Frage & Kodierung
zentrum	Studienzentrum 23=Dresden, 24=München

Bildung der Variable zentrum_r

Kodierungsbeschreibung:
zentrum_r wird auf 0 (Dresden) gesetzt, wenn zentrum den Wert 23 hat.
zentrum_r wird auf 1 (München) gesetzt, wenn zentrum den Wert 24 hat.

Kodierungsübersicht:

Verwendete Variablen	Abgeleitete Variablen
zentrum	**zentrum_r**
23	0
24	1

Code-Übersicht:

Variablenname	Frage & Kodierung
zentrum_r	Studienzentrum 0=Dresden, 1=München, NA=Missing

R-Code:

```
> zentrum_r <- NA
> zentrum_r[zentrum==23] <- 0
> zentrum_r[zentrum==24] <- 1
```

A.1.10 Geschwister

Die Variable siblings (Anzahl der Geschwister) aus ISAAC II wird zu einer dichotomen Variable GESCHW zusammengefasst, die angibt, ob die Person Geschwister hat (=1) oder nicht (=0).

Verwendete Variablen:

Variablenname	Frage & Kodierung
siblings	Anzahl Geschwister
	0-7=Anzahl der Geschwister, NA=Keine Angabe

Bildung der Variable GESCHW

Kodierungsbeschreibung:
GESCHW wird auf 0 gesetzt, wenn die Anzahl der Geschwister gleich 0 ist.
GESCHW wird auf 1 gesetzt, wenn 1-7 Geschwister angegeben wurden.

Kodierungsübersicht:

Verwendete Variablen	Abgeleitete Variablen
siblings	**GESCHW**
0	0
1-7	1

Code-Übersicht:

Variablenname	Frage & Kodierung
GESCHW	Geschwister vorhanden
	0=Nein, 1=Ja, NA=Missing

R-Code:

```
> GESCHW <- NA
> GESCHW[siblings==0] <- 0
> GESCHW[siblings!=0] <- 1
```

A.2 Variablen aus SOLAR I

A.2.1 Rauchverhalten

Die Angaben zum Rauchverhalten aus SOLAR I wurden so kodiert, dass man die Unterscheidung zwischen Raucher und Nichtraucher treffen kann.

Verwendete Variablen:

Variablenname	Frage & Kodierung
f54	Haben Sie selbst schon einmal Zigaretten geraucht?
	0=Nein, 1=Ja probiert, 2=Ja öfter NA=Keine Angabe
f55	Haben Sie schon einmal ein Jahr lang geraucht?
	0=Nein, 1=Ja, NA=Keine Angabe

In zwei Schritten, die im Folgenden ausführlich dargestellt werden, wurde aus den soeben aufgeführten Variablen die Variable RAUCHEN gebildet. Raucher ist hier jemand, der schon einmal ein Jahr lang geraucht hat. Nichtraucher ist somit jemand, der nicht bereits ein Jahr lang geraucht hat.

Gebildete Variable:

Variablenname	Frage & Kodierung
RAUCHEN	Rauchverhalten in SOLAR-I
	0=Nichtraucher, 1=Raucher, NA=Missing

Schritt 1: Bildung der Variable f55xx

Kodierungsbeschreibung:
Der Variablen f55xx werden zunächst die Werte aus f55 zugewiesen.
Ist die Variable f55 missing (NA) und hat f54 den Wert 0 oder 1, so wird der Variable f55xx der Wert 0 zugewiesen.
Ist die Variable f55 missing (NA) und hat f54 den Wert 2 oder ist missing (NA), so wird die Variable f55xx auf missing (NA) gesetzt.

Kodierungsübersicht:

Verwendete Variablen		Abgeleitete Variablen
f54	**f55**	**f55xx**
0, 1, 2, NA	0, 1	0, 1
0, 1	NA	0
2, NA	NA	NA

Code-Übersicht:

Variablenname	Frage & Kodierung
f55xx	Haben Sie schon einmal ein Jahr lang geraucht?
	0=Nein, 1=Ja, NA=Missing

R-Code:
Diese Variable wurde bereits vom Datenzentrum kodiert und war so im Datensatz enthalten, der als Grundlage für diese Arbeit galt. Eine selbstständige Kodierung mit Hilfe von R war daher nicht nötig.

Schritt 2: Bildung der Variable RAUCHEN

Kodierungsbeschreibung:
RAUCHEN wird auf 1 gesetzt, wenn f55xx den Wert 1 hat.
RAUCHEN wird auf missing (NA) gesetzt, wenn f55xx missing (NA) ist.
In allen anderen Fällen steht in der Variablen RAUCHEN der Wert 0.

Kodierungsübersicht:

Verwendete Variablen	**Abgeleitete Variablen**
f55xx	**RAUCHEN**
1	1
0	0
NA	NA

Code-Übersicht:

Variablenname	**Frage & Kodierung**
RAUCHEN	Rauchverhalten in SOLAR-I
	0=Nichtraucher, 1=Raucher, NA=Missing

R-Code:

```
> RAUCHEN <- 0
> RAUCHEN[!is.na(f55xx) & (f55xx==1)] <- 1
> RAUCHEN[is.na(f55xx)] <- NA
```

A.2.2 Berufssituation

Die Angaben zur Berufssituation aus SOLAR I lagen ursprünglich als mehrere Dummy-Variablen vor. Diese Variablen wurden so kodiert, dass sie als eine kategoriale Variable vorliegen. Im Rahmen dieser Kodierung wurden zusätzlich Doppelnennungen korrigiert, die in dieser Frage nicht erlaubt waren. Wie mit diesen Doppelnennungen umzugehen ist, wurde vorab gemeinsam mit Frau Kellberger und Herrn Heumann besprochen.

Verwendete Variablen:

Variablenname	**Frage & Kodierung**
f61_01xx	Sind Sie zur Zeit - HauptschülerIn 0=Nein, 1=Ja, NA=Missing
f61_02xx	Sind Sie zur Zeit - RealschülerIn 0=Nein, 1=Ja, NA=Missing
f61_03xx	Sind Sie zur Zeit - GymnasiastIn 0=Nein, 1=Ja, NA=Missing
f61_04xx	Sind Sie zur Zeit - SchülerIn einer anderen Schule 0=Nein, 1=Ja, NA=Missing
f61_05xx	Sind Sie zur Zeit - AuszubildendeR/BerufsschülerIn 0=Nein, 1=Ja, NA=Missing
f61_06xx	Sind Sie zur Zeit - StudentIn 0=Nein, 1=Ja, NA=Missing
f61_07xx	Sind Sie zur Zeit - angestellt 0=Nein, 1=Ja, NA=Missing
f61_08xx	Sind Sie zur Zeit - selbstständig 0=Nein, 1=Ja, NA=Missing
f61_09xx	Sind Sie zur Zeit - arbeitslos und arbeitssuchend 0=Nein, 1=Ja, NA=Missing
f61_10xx	Sind Sie zur Zeit - aus gesundheitl. Gründen nicht arbeitend 0=Nein, 1=Ja, NA=Missing
f61_11xx	Sind Sie zur Zeit - Hausfrau/Hausmann 0=Nein, 1=Ja, NA=Missing
f61_12xx	Sind Sie zur Zeit - sonstiges 0=Nein, 1=Ja, NA=Missing

Kommentar:
Die Variablen f61_01xx bis f61_12xx wurde bereits vom Datenzentrum aus den ursprünglichen Variablen f61_01 bis f61_12 kodiert und waren so im Datensatz enthalten, der als Grundlage für diese Arbeit galt. Wie bei der Kodierung vorgegangen wurde, zeigt folgende Tabelle.

Verwendete Variablen	**Abgeleitete Variablen**
f61_01 - f61_12	**f61_01xx - f61_12xx**
1	1
NA, aber nicht alle 12 Variablen NA	0
alle 12 Variablen NA	NA

Bildung der Variable BERUF

Kodierungsbeschreibung:
Hat die dichotome Variable f61_01xx den Wert 1, so wird die Variable BERUF auf 1 gesetzt. Hat die dichotome Variable f61_02xx den Wert 1, so wird die Variable BERUF auf 2 gesetzt. Nach diesem Schema wird für alle Variablen vorgegangen.
In einigen Fällen nimmt mehr als eine dichotome Variable den Wert 1 an. Wie diese Sonderfälle kodiert wurden, kann folgender Tabelle entnommen werden.

Kodierungsübersicht - Sonderfälle:

Variablen	Zugewiesener Wert für BERUF
f61_01xx = 1 und f61_05xx = 1	5
f61_02xx = 1 und f61_12xx = 1	2
f61_03xx = 1 und f61_12xx = 1	3
f61_03xx = 1 und f61_07xx = 1	3
f61_03xx = 1 und f61_04xx = 1	3
f61_04xx = 1 und f61_09xx = 1	4
f61_05xx = 1 und f61_12xx = 1	5
f61_03xx = 1 und f61_10xx = 1	5
f61_03xx = 1 und f61_07xx = 1	5
f61_05xx = 1 und f61_12xx = 1	5

Code-Übersicht:

Variablenname	Frage & Kodierung
BERUF	Berufssituation - SOLAR I
	1=HauptschülerIn
	2=RealschülerIn
	3=GymnasiastIn
	4=SchülerIn einer anderen Schule
	5=AuszubildendeR/BerufsschülerIn
	6=StudentIn
	7=Angestellt
	8=Selbstständig
	9=Arbeitslos und arbeitssuchend
	10=Aus gesundheitl. Gründen nicht arbeitend
	11=Hausfrau/Hausmann
	12=Sonstiges
	NA=Missing

R-Code:

```
> BERUF<-NA
> BERUF[!is.na(f61_01xx) & (f61_01xx==1)] <- 1
> BERUF[!is.na(f61_02xx) & (f61_02xx==1)] <- 2
> BERUF[!is.na(f61_03xx) & (f61_03xx==1)] <- 3
> BERUF[!is.na(f61_04xx) & (f61_04xx==1)] <- 4
> BERUF[!is.na(f61_05xx) & (f61_05xx==1)] <- 5
> BERUF[!is.na(f61_06xx) & (f61_06xx==1)] <- 6
> BERUF[!is.na(f61_07xx) & (f61_07xx==1)] <- 7
> BERUF[!is.na(f61_08xx) & (f61_08xx==1)] <- 8
> BERUF[!is.na(f61_09xx) & (f61_09xx==1)] <- 9
> BERUF[!is.na(f61_10xx) & (f61_10xx==1)] <- 10
> BERUF[!is.na(f61_11xx) & (f61_11xx==1)] <- 11
> BERUF[!is.na(f61_12xx) & (f61_12xx==1)] <- 12
> BERUF[!is.na(f61_01xx) & (f61_01xx==1) & is.na(f61_05xx) & (f61_05xx==1)] <- 5
> BERUF[!is.na(f61_02xx) & (f61_02xx==1) & !is.na(f61_12xx) & (f61_12xx==1)] <- 2
> BERUF[!is.na(f61_03xx) & (f61_03xx==1) & !is.na(f61_12xx) & (f61_12xx==1)] <- 3
> BERUF[!is.na(f61_03xx) & (f61_03xx==1) & !is.na(f61_07xx) & (f61_07xx==1)] <- 3
> BERUF[!is.na(f61_03xx) & (f61_03xx==1) & !is.na(f61_04xx) & (f61_04xx==1)] <- 3
> BERUF[!is.na(f61_04xx) & (f61_04xx==1) & !is.na(f61_09xx) & (f61_09xx==1)] <- 4
> BERUF[!is.na(f61_05xx) & (f61_05xx==1) & !is.na(f61_12xx) & (f61_12xx==1)] <- 5
> BERUF[!is.na(f61_05xx) & (f61_05xx==1) & !is.na(f61_10xx) & (f61_10xx==1)] <- 5
> BERUF[!is.na(f61_05xx) & (f61_05xx==1) & !is.na(f61_07xx) & (f61_07xx==1)] <- 5
> BERUF[!is.na(f61_05xx) & (f61_05xx==1) & !is.na(f61_12xx) & (f61_12xx==1)] <- 5
```

A.3 Variablen aus SOLAR II

A.3.1 Asthma

Diese Kodierung für SOLAR II entspricht der Kodierung der Variable CURASTHV in SOLAR I.

Verwendete Variablen:

Variablenname	Frage & Kodierung
s2f07	Haben Sie jemals in den letzten 12 Monaten ein pfeifendes oder brummendes Geräusch in Ihrem Brustkorb gehört? 0=Nein, 1=Ja, NA=Keine Angabe
s2f19	Haben Sie jemals Asthma gehabt? 0=Nein, 1=Ja, NA=Keine Angabe
s2f20_01	Wurde bei Ihnen von einem Arzt schon einmal eine der folgenden Erkrankungen festgestellt? Asthma 0=Noch nie, 1=Einmal, 2=Mehrmals, NA=Keine Angabe
s2f20_02	Wurde bei Ihnen von einem Arzt schon einmal eine der folgenden Erkrankungen festgestellt? Spastische/asthmatische Bronchitis 0=Noch nie, 1=Einmal, 2=Mehrmals, NA=Keine Angabe

In fünf Schritten, die im Folgenden ausführlich dargestellt werden, wurde aus den soeben aufgeführten Variablen die Variable s2CURASTHV gebildet. Diese Variable zeigt an, ob bei einem Probanden derzeit Asthma vorliegt oder nicht. Asthma liegt hier bei einem Probanden vor, wenn diese Person bei sich selbst Asthmasymptome (pfeifendes oder brummendes Geräusch im Brustkorb) innerhalb der letzten 12 Monate beobachtet hat und gleichzeitig eine Arztdiagnose Asthma oder spastische/asthmatische Bronchitis vorliegt (d.h. Asthma bereits einmal oder mehrmals oder spastische/asthmatische Bronchitis mehrmals von einem Arzt diagnostiziert).

Gebildete Variable:

Variablenname	Frage & Kodierung
s2CURASTHV	Current Asthma (derzeit Asthma) 0=Nein, 1=Ja, NA=Missing

Schritt 1: Bildung der Variablen s2f20_01x und s2f20_02x

Kodierungsbeschreibung:
Den Variablen s2f20_01x und s2f20_02x werden zunächst die Werte aus s2f20_01 bzw. s2f20_02 zugewiesen. Wurde Frage 19 mit 0 oder 1 beantwortet und wurde in Frage 20 auf mind. eine der Variablen s2f20_01 oder s2f20_02 keine Angabe gegeben, so wird die jeweilige Variable auf missing (NA) gesetzt. Fehlen die Angaben zu Frage 19 und 20 komplett, so werden die Variablen s2f20_01x und s2f20_02x beide auf missing (NA) gesetzt. In allen anderen Fällen stimmen die Werte von s2f20_01x und s2f20_02x mit den Werten aus s2f20_01 und s2f20_02 überein.

Kodierungsübersicht:

Verwendete Variablen			Abgeleitete Variablen	
s2f19	**s2f20_01**	**s2f20_02**	**s2f20_01x**	**s2f20_02x**
0, 1, NA	0, 1, 2	0, 1, 2	0, 1, 2	0, 1, 2
0, 1	0, 1, 2	NA	0, 1, 2	NA
0, 1	NA	0, 1, 2	NA	0, 1, 2
0, 1	NA	NA	NA	NA
NA	NA	NA	NA	NA

Code-Übersicht:

Variablenname	Frage & Kodierung
s2f20_01x	Asthma 0=Noch nie, 1=Einmal, 2=Mehrmals, NA=Missing
s2f20_02x	Spastische/asthmatische Bronchitis 0=Noch nie, 1=Einmal, 2=Mehrmals, NA=Missing

R-Code:

Diese Variable wurde bereits vom Datenzentrum kodiert und war so im Datensatz enthalten, der als Grundlage für diese Arbeit galt. Eine selbstständige Kodierung mit Hilfe von R war daher nicht nötig.

Schritt 2: Bildung der Hilfsvariablen s2f20_01n und s2f20_02n

Kodierungsbeschreibung:
Den Hilfsvariablen werden zunächst die Werte aus s2f20_01x bzw. s2f20_02x zugewiesen. Ist in Frage 20 eine der Variablen s2f20_01x und s2f20_02x mit 0, 1 oder 2 beantwortet, so gilt die komplette Frage als beantwortet. Für diesen Fall werden die Hilfsvariablen s2f20_01n und s2f20_02n von missing (NA) auf 0 gesetzt. Man nimmt also in diesen Fällen an, dass die jeweilige Erkrankung noch nie von einem Arzt festgestellt wurde. In allen anderen Fällen stimmen die Werte von s2f20_01n und s2f20_02n mit den Werten aus s2f20_01x und s2f20_02x überein.

Kodierungsübersicht:

Verwendete Variablen		**Abgeleitete Variablen**	
s2f20_01x	**s2f20_02x**	**s2f20_01n**	**s2f20_02n**
0, 1, 2	0, 1, 2	0, 1, 2	0, 1, 2
0, 1, 2	NA	0, 1, 2	0
NA	0, 1, 2	0	0, 1, 2
NA	NA	NA	NA

Code-Übersicht:

Variablenname	**Frage & Kodierung**
s2f20_01n	Asthma 0=Noch nie, 1=Einmal, 2=Mehrmals, NA=Missing
s2f20_02n	Spastische/asthmatische Bronchitis 0=Noch nie, 1=Einmal, 2=Mehrmals, NA=Missing

R-Code:

```
> s2f20_01n <- s2f20_01x
> s2f20_02n <- s2f20_02x
> s2f20_01n[is.na(s2f20_01x) & !is.na(s2f20_02x)] <- 0
> s2f20_02n[is.na(s2f20_02x) & !is.na(s2f20_01x)] <- 0
```

Schritt 3: Bildung der Variable s2ARASOBS

Kodierungsbeschreibung:
s2ARASOBS wird auf 1 gesetzt, wenn s2f20_01n den Wert 1 oder 2 oder s2f20_02n den Wert 2 hat (d.h. Asthma bereits einmal oder mehrmals oder spastische/asthmatische Bronchitis mehrmals von einem Arzt festgestellt).
s2ARASOBS wird auf 0 gesetzt, wenn s2f20_01n den Wert 0 und s2f20_02n den Wert 1 oder 0 hat.
In allen anderen Fällen ist s2ARASOBS missing (NA).

Kodierungsübersicht:

Verwendete Variablen		**Abgeleitete Variablen**
s2f20_01n	**s2f20_02n**	**s2ARASOBS**
1, 2	0, 1, 2	1
0, 1, 2	2	1
0	0,1	0
NA	NA	NA

Code-Übersicht:

Variablenname	**Frage & Kodierung**
s2ARASOBS	Arztdiagnose Asthma oder spastische/asthmatische Bronchitis 0=Nein, 1=Ja, NA=Missing

R-Code:

```
> s2ARASOBS <- NA
> s2ARASOBS[!is.na(s2f20_01n) & (s2f20_01n==1)] <- 1
> s2ARASOBS[!is.na(s2f20_01n) & (s2f20_01n==2)] <- 1
> s2ARASOBS[!is.na(s2f20_02n) & (s2f20_02n==2)] <- 1
> s2ARASOBS[!is.na(s2f20_01n) & (s2f20_01n==0)
+ & !is.na(s2f20_02n) & (s2f20_02n==1)] <- 0
> s2ARASOBS[!is.na(s2f20_01n) & (s2f20_01n==0)
+ & !is.na(s2f20_02n) & (s2f20_02n==0)] <- 0
```

Schritt 4: Bildung der Variable s2KEUASOBV

Kodierungsbeschreibung:
In dieser Variable werden die Informationen über das Vorliegen eines Asthmasymptoms aus den Fragen 7 und 20 kombiniert.
s2KEUASOBV wird auf 1 gesetzt, wenn s2f07 und s2ARASOBS beiden den Wert 1 haben (d.h. Arztdiagnose Asthma oder spastische/asthmatische Bronchitis und gleichzeitig Asthmasymptome bei sich selbst beobachtet).
s2KEUASOBV wird auf 3 gesetzt, wenn s2f07 und s2ARASOBS beiden den Wert 0 haben (d.h. weder Arztdiagnose, noch Symptome bei sich selbst beobachtet).
s2KEUASOBV wird auf 2 gesetzt, wenn nur zu einer der beiden Variablen s2f07 und s2ARASOBS eine Angabe vorliegt oder wenn sich die Angaben unterscheiden.
s2KEUASOBV ist nur dann missing (NA), wenn weder zu s2f07 noch zu s2ARASOBS Angaben vorliegen.

Kodierungsübersicht:

Verwendete Variablen		**Abgeleitete Variablen**
s2f07	**s2ARASOBS**	**s2KEUASOBV**
1	1	1
0	0	3
1	0	2
0	1	2
1, 0	NA	2
NA	0, 1	2
NA	NA	NA

Code-Übersicht:

Variablenname	**Frage & Kodierung**
s2KEUASOBV	Wheezing und ARASOBS Asthmasymptome bei sich selbst beobachtet und gleichzeitig Arztdiagnose Asthma oder spastische/asthmatische Bronchitis 1=Positiv, 2=Intermediate, 3=Negativ, NA=Missing

R-Code:

```
> s2KEUASOBV <- 2
> s2KEUASOBV[!is.na(s2f07) & (s2f07==1)
+ & !is.na(s2ARASOBS) & (s2ARASOBS==1)] <- 1
> s2KEUASOBV[!is.na(s2f07) & (s2f07==0)
+ & !is.na(s2ARASOBS) & (s2ARASOBS==0)] <- 3
> s2KEUASOBV[is.na(s2f07) & is.na(s2ARASOBS)] <- NA
```

Schritt 5: Bildung der Variablen s2CURASTHV

Kodierungsbeschreibung:
s2CURASTHV wird auf 1 gesetzt, wenn s2KEUASOBV mit 1 (positiv) kodiert wurde.
s2CURASTHV ist missing (NA), wenn s2KEUASOBV missing (NA) ist.
In allen anderen Fällen (s2KEUASOBV = 2 oder s2KEUASOBV = 3) steht in s2CURASTHV der Wert 0.

Kodierungsübersicht:

Verwendete Variablen	**Abgeleitete Variablen**
s2KEUASOBV	**s2CURASTHV**
1	1
2, 3	0
NA	NA

Code-Übersicht:

Variablenname	**Frage & Kodierung**
s2CURASTHV	Current Asthma (derzeit Asthma) 0=Nein, 1=Ja, NA=Missing

R-Code:

```
> s2CURASTHV <- 0
> s2CURASTHV[!is.na(s2KEUASOBV) & (s2KEUASOBV==1)] <- 1
> s2CURASTHV[is.na(s2KEUASOBV)] <- NA
```

A.3.2 Allergische Rhinitis

Diese Kodierung für SOLAR II entspricht der Kodierung der Variable CURHAYV in SOLAR I.

Verwendete Variablen:

Variablenname	Frage & Kodierung
s2f26	Hatten Sie in den letzten 12 Monaten Probleme mit Niesanfällen oder einer laufenden, verstopften Nase, ohne erkältet zu sein? 0=Nein, 1=Ja, NA=Keine Angabe
s2f27	Traten diese Nasenprobleme zusammen mit juckenden, tränenden Augen auf? 0=Nein, 1=Ja, NA=Keine Angabe
s2f30	Hatten Sie in den letzten 12 Monaten allergischen Schupfen, z.B. “Heuschnupfen”? 0=Nein, 1=Ja, NA=Keine Angabe
s2f33	Hat ein Arzt bei Ihnen schon einmal allergischen Schnupfen, zum Beispiel “Heuschnupfen” festgestellt? 0=Nein, 1=Ja, NA=Keine Angabe

In drei Schritten, die im Folgenden ausführlich dargestellt werden, wurde aus den soeben aufgeführten Variablen die Variable s2CURHAYV gebildet. Diese Variable zeigt an, ob bei einem Probanden derzeit Allergische Rhinitis vorliegt oder nicht. Allergische Rhinitis liegt hier bei einem Probanden vor, wenn bei dieser Person innerhalb der letzten 12 Monate Nasenprobleme (Niesanfälle oder laufende, verstopfte Nase ohne Erkältung) zusammen mit juckenden, tränenden Augen auftraten und gleichzeitig schon einmal von einem Arzt allergischer Schnupfen diagnostiziert wurde.

Gebildete Variable

Variablenname	Frage & Kodierung
s2CURHAYV	Current Hayfever (derzeit Allergische Rhinitis) 0=Nein, 1=Ja, NA=Missing

Schritt 1: Bildung der Variable s2f27xx

Kodierungsbeschreibung:
Der Variablen s2f27xx werden zunächst die Werte aus s2f27 zugewiesen. Ist die Variable s2f27 missing (NA) und hat s2f26 den Wert 0, so hat s2f27xx ebenfalls den Wert 0. Ist die Variable s2f27 missing (NA) und hat s2f26 den Wert 1 oder ist missing (NA), so wird die Variable s2f27xx auf missing (NA) gesetzt.

Kodierungsübersicht:

Verwendete Variablen		Abgeleitete Variablen
s2f26	**s2f27**	**s2f27xx**
0, 1, NA	0, 1	0, 1
0	NA	0
1, NA	NA	NA

Code-Übersicht:

Variablenname	Frage & Kodierung
s2f27xx	Traten diese Nasenprobleme zusammen mit juckenden, tränenden Augen auf? 0=Nein, 1=Ja, NA=Missing

R-Code:

```
> s2f27xx <- s2f27
> s2f27xx[!is.na(s2f26) & (s2f26==0) & is.na(s2f27)] <- 0
> s2f27xx[!is.na(s2f26) & (s2f26==1) & is.na(s2f27)] <- NA
> s2f27xx[is.na(s2f26) & is.na(s2f27)] <- NA
```

Schritt 2: Bildung der Variable s2f33xx

Kodierungsbeschreibung:
Der Variablen s2f33xx werden zunächst die Werte aus s2f33 zugewiesen. Ist die Variable s2f33 missing (NA) und hat s2f30 den Wert 0, so hat s2f33xx ebenfalls den Wert 0. Ist die Variable s2f33 missing (NA) und hat s2f30 den Wert 1 oder ist missing (NA), so wird die Variable s2f33xx auf missing (NA) gesetzt.

Kodierungsübersicht:

Verwendete Variablen		Abgeleitete Variablen
s2f30	**s2f33**	**s2f33xx**
0, 1, NA	0, 1	0, 1
0	NA	0
1, NA	NA	NA

Code-Übersicht:

Variablenname	Frage & Kodierung
s2f33xx	Hat ein Arzt bei Ihnen schon einmal allergischen Schnupfen, zum Beispiel “Heuschnupfen” festgestellt? 0=Nein, 1=Ja, NA=Missing

R-Code:

```
> s2f33xx <- s2f33
> s2f33xx[!is.na(s2f30) & (s2f30==0) & is.na(s2f33)] <- 0
> s2f33xx[!is.na(s2f30) & (s2f30==1) & is.na(s2f33)] <- NA
> s2f33xx[is.na(s2f30) & is.na(s2f33)] <- NA
```

Schritt 3: Bildung der Variable s2CURHAYV

Kodierungsbeschreibung:
s2CURHAYV wird auf 1 gesetzt, wenn s2f27xx und s2f33xx beide den Wert 1 annehmen. s2CURHAYV wird auf missing (NA) gesetzt, wenn s2f27xx und s2f33xx beide missing (NA) sind. In allen anderen Fällen steht in der Variablen s2CURHAYV der Wert 0.

Kodierungsübersicht:

Verwendete Variablen		Abgeleitete Variablen
s2f27xx	**s2f33xx**	**s2CURHAYV**
1	1	1
1,0	0,NA	0
0, NA	1,0	0
NA	NA	NA

Code-Übersicht:

Variablenname	Frage & Kodierung
s2CURHAYV	Current Hayfever (derzeit Allergische Rhinitis) 0=Nein, 1=Ja, NA=Missing

R-Code:

```
> s2CURHAYV <- 0
> s2CURHAYV[!is.na(s2f27xx) & (s2f27xx==1)
+ & !is.na(s2f33xx) & (s2f33xx==1)] <- 1
> s2CURHAYV[is.na(s2f27xx) & is.na(s2f33xx)] <- NA
```

A.3.3 Rauchverhalten

Die Angaben zum Rauchverhalten aus SOLAR II wurden so kodiert, dass man die Unterscheidung zwischen Raucher, Ex-Raucher und Nichtraucher treffen kann. Nichtraucher werden analog zu SOLAR I definiert. In SOLAR II trifft man (im Vergleich zu SOLAR I) noch zusätzlich die Unterscheidung zwischen Raucher und Ex-Raucher, da man davon ausgehen kann, dass sich bei den Probanden das Rauchverhalten im Gegensatz zum Zeitpunkt der SOLAR I-Studie nun weitestgehend stabilisiert hat. Raucher sind Personen, die schon einmal ein Jahr lang geraucht haben und dies auch innerhalb des letzten Monats getan haben. Ex-Raucher sind Personen, die zwar schon einmal ein Jahr lang geraucht haben, aber dies nicht innerhalb des letzten Monats getan haben.

Verwendete Variablen:

Variablenname	Frage & Kodierung
s2f73	Haben Sie schon einmal ein Jahr lang geraucht? 0=Nein, 1=Ja, NA=Keine Angabe
s2f75	Haben Sie innerhalb des letzten Monats geraucht? 0=Nein, 1=Ja, NA=Keine Angabe

Bildung der Variable s2RAUCHEN

Kodierungsbeschreibung:
s2RAUCHEN wird auf 0 (Nichtraucher) gesetzt, wenn s2f73 den Wert 0 hat.
s2RAUCHEN wird auf 2 (Ex-Raucher) gesetzt, wenn s2f73 den Wert 1 hat und s2f75 den Wert 0 hat.
s2RAUCHEN wird auf missing (NA) gesetzt, wenn sowohl s2f73 als auch s2f75 missing (NA) sind.
In allen anderen Fällen steht in der Variablen s2RAUCHEN der Wert 1 (Raucher).

Kodierungsübersicht:

Verwendete Variablen		Abgeleitete Variablen
s2f73	**s2f75**	**s2RAUCHEN**
0	1,0	0
1	1	1
NA	1	1
1	NA	1
1	0	2
NA	NA	NA

Code-Übersicht:

Variablenname	Frage & Kodierung
s2RAUCHEN	Rauchverhalten in SOLAR-II 0=Nichtraucher, 1=Raucher, 2=Ex-Raucher, NA=Missing

R-Code:

```
> s2RAUCHEN <- 1
> s2RAUCHEN[is.na(s2f73) & is.na(s2f75)] <- NA
> s2RAUCHEN[!is.na(s2f73) & (s2f73==1)
+ & !is.na(s2f75) & (s2f75==0)] <- 2
> s2RAUCHEN[!is.na(s2f73) & (s2f73==0)] <- 0
```

A.3.4 Berufssituation

Die Angaben zur Berufssituation aus SOLAR II lagen ursprünglich als mehrere Dummy-Variablen vor. Diese Variablen wurden so kodiert, dass sie als eine kategoriale Variable vorliegen. Im Rahmen dieser Kodierung wurden zusätzlich Doppelnennungen korrigiert, die in dieser Frage nicht erlaubt waren. Wie mit diesen Doppelnennungen umzugehen ist, wurde vorab gemeinsam mit Frau Kellberger und Herrn Heumann besprochen.

Verwendete Variablen:

Variablenname	Frage & Kodierung
s2f81_01	Sind Sie zur Zeit - AuszubildendeR/BerufsschülerIn 0=Nein, 1=Ja, NA=Missing
s2f81_02	Sind Sie zur Zeit - StudentIn (hauptberuflich) 0=Nein, 1=Ja, NA=Missing
s2f81_03	Sind Sie zur Zeit - Angestellt 0=Nein, 1=Ja, NA=Missing
s2f81_04	Sind Sie zur Zeit - Selbstständig 0=Nein, 1=Ja, NA=Missing
s2f81_05	Sind Sie zur Zeit - Arbeitslos und arbeitssuchend 0=Nein, 1=Ja, NA=Missing
s2f81_06	Sind Sie zur Zeit - Aus gesundheitl. Gründen nicht arbeitend 0=Nein, 1=Ja, NA=Missing
s2f81_07	Sind Sie zur Zeit - Hausmann/Hausfrau (hauptberuflich) 0=Nein, 1=Ja, NA=Missing
s2f81_08	Sind Sie zur Zeit - In Mutterschutz / Elternzeit oder sonstige Beurlaubung 0=Nein, 1=Ja, NA=Missing
s2f81_09	Sind Sie zur Zeit - Sonstiges 0=Nein, 1=Ja, NA=Missing

Bildung der Variable s2BERUF

Kodierungsbeschreibung:
Hat die dichotome Variable s2f81_01 den Wert 1, so wird die Variable s2BERUF auf 1 gesetzt. Hat die dichotome Variable s2f81_02 den Wert 1, so wird die Variable s2BERUF auf 2 gesetzt. Nach diesem Schema wird für alle Variablen vorgegangen.
In einigen Fällen nimmt mehr als eine dichotome Variable den Wert 1 an. Wie diese Sonderfälle kodiert wurden, kann folgender Tabelle entnommen werden.

Kodierungsübersicht - Sonderfälle:

Variablen	Zugewiesener Wert für s2BERUF
s2f81_05 = 1 und s2f81_09 = 1 (Nebenjob)	3
s2f81_03 = 1 und s2f81_09 = 1	3
s2f81_02 = 1 und s2f81_09 = 1	2
s2f81_02 = 1 und s2f81_03 = 1	2
s2f81_02 = 1 und s2f81_04 = 1	2
s2f81_01 = 1 und s2f81_09 = 1	1
s2f81_01 = 1 und s2f81_08 = 1	1
s2f81_01 = 1 und s2f81_05 = 1	1
s2f81_01 = 1 und s2f81_04 = 1	1
s2f81_01 = 1 und s2f81_03 = 1	1

Code-Übersicht:

Variablenname	Frage & Kodierung
s2BERUF	Berufssituation - SOLAR-II 1=AuszubildendeR/BerufsschülerIn 2=StudentIn (hauptberuflich) 3=Angestellt 4=Selbstständig 5=Arbeitslos und arbeitssuchend 6=Aus gesundheitl. Gründen nicht arbeitend 7=Hausfrau/Hausmann (hauptberuflich) 8=In Mutterschutz / Elternzeit oder sonstige Beurlaubung 9=Sonstiges

R-Code:

```
> s2BERUF<-NA
> s2BERUF[!is.na(s2f81_01) & (s2f81_01==1)] <- 1
> s2BERUF[!is.na(s2f81_02) & (s2f81_02==1)] <- 2
> s2BERUF[!is.na(s2f81_03) & (s2f81_03==1)] <- 3
> s2BERUF[!is.na(s2f81_04) & (s2f81_04==1)] <- 4
> s2BERUF[!is.na(s2f81_05) & (s2f81_05==1)] <- 5
> s2BERUF[!is.na(s2f81_06) & (s2f81_06==1)] <- 6
> s2BERUF[!is.na(s2f81_07) & (s2f81_07==1)] <- 7
> s2BERUF[!is.na(s2f81_08) & (s2f81_08==1)] <- 8
> s2BERUF[!is.na(s2f81_09) & (s2f81_09==1)] <- 9
> s2BERUF[!is.na(s2f81_05) & (s2f81_05==1) & !is.na(s2f81_09) & (s2f81_09==1)] <- 3
> s2BERUF[!is.na(s2f81_03) & (s2f81_03==1) & !is.na(s2f81_09) & (s2f81_09==1)] <- 3
> s2BERUF[!is.na(s2f81_02) & (s2f81_02==1) & !is.na(s2f81_09) & (s2f81_09==1)] <- 2
> s2BERUF[!is.na(s2f81_02) & (s2f81_02==1) & !is.na(s2f81_03) & (s2f81_03==1)] <- 2
> s2BERUF[!is.na(s2f81_02) & (s2f81_02==1) & !is.na(s2f81_04) & (s2f81_04==1)] <- 2
> s2BERUF[!is.na(s2f81_01) & (s2f81_01==1) & !is.na(s2f81_09) & (s2f81_09==1)] <- 1
> s2BERUF[!is.na(s2f81_01) & (s2f81_01==1) & !is.na(s2f81_08) & (s2f81_08==1)] <- 1
> s2BERUF[!is.na(s2f81_01) & (s2f81_01==1) & !is.na(s2f81_05) & (s2f81_05==1)] <- 1
> s2BERUF[!is.na(s2f81_01) & (s2f81_01==1) & !is.na(s2f81_04) & (s2f81_04==1)] <- 1
> s2BERUF[!is.na(s2f81_01) & (s2f81_01==1) & !is.na(s2f81_03) & (s2f81_03==1)] <- 1
```

A.3.5 Schulbildung

Die Angaben zur Schulbildung aus SOLAR II wurden so kodiert, dass man die Unterscheidung zwischen höherere und niedrigerer Schulbildung treffen kann. "Höhere Schulbildung" hat ein Proband, wenn er als höchsten Schulabschluss Fachhochschule, fachgebundene Hochschulreife, Abitur oder allgemeine Hochschulreife angab. Bei allen anderen Angaben wurde ihm "niedrigere Schulbildung" zugeordnet.

Verwendete Variablen:

Variablenname	Frage & Kodierung
s2f80	Welchen Schulabschluss haben Sie? Wenn Sie mehrere Abschlüsse haben, nennen Sie nur den höchsten! 0=Hauptschulabschluss/Volksschulabschluss(Mittelschule) 1=Realschulabschluss(mittlere Reife, Mittelschule) 2=Fachhochschulreife/fachgebundene Hochschulreife 3=Abitur/allgemeine Hochschulreife 4=Anderen Schulabschluss 5=Schule beendet ohne Abschluss 6=Noch keinen Schulabschluss

Bildung der Variable s2SCHULE

Kodierungsbeschreibung:
s2SCHULE wird auf 1 (höhere Schulbildung) gesetzt, wenn s2f80 den Wert 2 (FH) oder 3 (Abitur) hat.
s2SCHULE wird auf missing (NA) gesetzt, wenn s2f80 missing (NA) ist.
In allen anderen Fällen steht in der Variablen s2SCHULE der Wert 0 (niedrigere Schulbildung).

Kodierungsübersicht:

Variablen	Zugewiesener Wert für s2SCHULE
s2f80	**s2SCHULE**
2,3	1
0,1,4,5,6	0
NA	NA

Code-Übersicht:

Variablenname	Frage & Kodierung
s2SCHULE	Schulbildung in SOLAR II 0=Niedrigere Schulbildung, 1=Höhere Schulbildung, NA=Missing

R-Code:

```
> s2SCHULE <- 0
> s2SCHULE[is.na(s2f80)] <- NA
> s2SCHULE[!is.na(s2f80) & (s2f80==2)] <- 1
> s2SCHULE[!is.na(s2f80) & (s2f80==3)] <- 1
```

A.4 Benötigte Variablen für die Tätigkeitsdaten

Zu den ausgeführten Tätigkeiten wurden in SOLAR I und SOLAR II jeweils zwei Fragen gestellt. Alle im weiteren durchgeführten Rekodierungen basieren auf diesen Fragen.

Verwendete Fragen aus SOLAR I:

Fragennummer	**Frage & Kodierung**
Frage 65	Haben Sie schon einmal irgendeine Arbeit / irgendeinen Ferienjob gehabt? 0=Nein, 1=Ja, NA=Missing
Frage 66	Welche Art von Arbeitsstellen und/oder Ferienjobs etc. hatten Sie bis jetzt (mind. 1 Monat lang, mind. 8 Stunden pro Woche)? offene Angaben zu Tätigkeit, Branche, Beginn und Ende der Tätigkeit sowie zu den Stunden pro Woche

Verwendete Fragen aus SOLAR II:

Fragennummer	**Frage & Kodierung**
Frage 92	Haben Sie seit der letzten SOLAR-Studie (2003/2004) irgendeine Arbeit/irgendeinen Ferienjob für mindestens 1 Monat gehabt? 0=Nein, 1=Ja, NA=Missing
Frage 93	Welche Art von Arbeitsstellen und/oder Ferienjobs etc. hatten Sie seit der letzten SOLAR-Studie (2003/2004) (mind. 1 Monat lang, mind. 8 Stunden pro Woche)? offene Angaben zu Tätigkeit, Branche, Beginn und Ende der Tätigkeit sowie zu den Stunden pro Woche

A.4.1 Gearbeitet in SOLAR I

Eine Person, die mind. einen Eintrag bei den Tätigkeitsdaten gemacht hatte (Frage 66) muss auch die vorherigen Frage (Frage 65) bejahen, ob in diesem Zeitraum gearbeitet wurde. Bei den Probanden, bei denen das nicht der Fall war, wurde die Variable f65x in einer neuen Variable f65xx entsprechend korrigiert. Diese Variable gibt also an, ob überhaupt gearbeitet wurde, unabhängig davon, wie viele Wochenstunden gearbeitet wurden.

Bildung der Variablen f65xx und GEARB_s1 (pro Proband)

Kodierungsbeschreibung:
Der Variablen f65xx werden zunächst die Werte aus f65x zugewiesen. Wurde mind. eine Tätigkeitsangaben in Frage 66 getätigt (d.h. Variable n_jobs ≥ 1), so wird die Variable f65xx auf 1 gesetzt (wenn sie zuvor nicht bereits den Wert 1 hatte). Alle Werte aus f65xx werden dann in die Variable GEARB_s1 kopiert.

Kodierungsübersicht:

Verwendete Variablen		**Abgeleitete Variablen**	
f65x	**Frage66**	**f65xx**	**GEARB_s1**
0	NA (d.h. n_jobs = 0)	0	0
0	Angaben (d.h. n_jobs ≥ 1)	1	1
1	Angaben (d.h. n_jobs ≥ 1)	1	1
NA	Angaben, NA	NA	NA

Code-Übersicht:

Variablenname	**Frage & Kodierung**
f65xx	Haben Sie schon einmal irgendeine Arbeit / irgendeinen Ferienjob gehabt? 0=Nein, 1=Ja, NA=Missing
GEARB_s1	Haben Sie schon einmal irgendeine Arbeit / irgendeinen Ferienjob gehabt? 0=Nein, 1=Ja, NA=Missing

R-Code:

```
> f65xx <- f65x
> f65xx[(n_jobs==1) & (f65x==0)] <- 1
Die Übertragung der Variable f65xx in die Variable GEARB\_s1 erfolgte
im Rahmen der Datensatzerstellung.
```

A.4.2 Gearbeitet in SOLAR II

Eine Person, die mind. einen Eintrag bei den Tätigkeitsdaten gemacht hatte (Frage 93) muss auch die vorherigen Frage (Frage 92) bejahen, ob in diesem Zeitraum gearbeitet wurde. Bei den Probanden, bei denen das nicht der Fall war, wurde die Variable s2f92 in einer neuen Variable GEARB_s2 entsprechend korrigiert. Diese Variable gibt also an, ob überhaupt gearbeitet wurde, unabhängig davon, wie viele Wochenstunden gearbeitet wurden.

Bildung der Variable GEARB_s2 (pro Proband)

Kodierungsbeschreibung:
Der Variable GEARB_s2 werden zunächst die Werte aus s2f92 zugewiesen. Wurde mind. eine Tätigkeitsangabe in Frage 93 getätigt, so wird die Variable GEARB_s2 auf 1 gesetzt (wenn sie zuvor nicht bereits den Wert 1 hatte). Wurde in Frage 93 die Antwortoption "Keine Tätigkeit für mind. 8 Stunden pro Woche ausgeführt" genutzt (s2f93_01 = 1), so wurde GEARB_s2 ebenfalls mit 1 kodiert.

Kodierungsübersicht:

Verwendete Variablen		**Abgeleitete Variablen**
s2f92	**Frage93**	**GEARB_s2**
0	NA	0
0	Angaben	1
1	Angaben	1
NA	Angaben, NA	NA
1	s2f93_01 = 1	1

Code-Übersicht:

Variablenname	**Frage & Kodierung**
GEARB_s2	Haben Sie seit der letzten SOLAR-Studie (2003/2004) irgendeine Arbeit/irgendeinen Ferienjob für mindestens 1 Monat gehabt? 0=Nein, 1=Ja, NA=Missing

R-Code:

```
> GEARB_s2 <- s2f92
> GEARB_s2[(knr=="A59355296")|(knr=="B56628279")|(knr=="B58427289")|(knr=="B59327287")|(knr=="D56158285")
+ |(knr=="K54357297")|(knr=="N53657299")|(knr=="N55156285")|(knr=="N58928280")|(knr=="N59156296")
+ |(knr=="P53858298")|(knr=="P56028289")|(knr=="P59827272")|(knr=="R50927282")|(knr=="R59827287")
+ |(knr=="S52627272")|(knr=="S53151284")|(knr=="T54857294")|(knr=="T56158280")|(knr=="T57626276")
+ |(knr=="U56726296")|(knr=="U57955278")|(knr=="U59828294")|(knr=="D54726284")|(knr=="P56957272")
+ |(knr=="P58027298")|(knr=="U53256284")] <- 1
> GEARB_s2[s2f93_01==1] <- 1
```

A.4.3 Gearbeitet in SOLAR I und/oder SOLAR II (unabhängig von der Anzahl der Wochenstunden)

Aus den beiden separaten Variablen GEARB_s1 und GEARB_s2 wird im Folgenden die Variable GEARB_s1s2 gebildet, die angibt, ob der Proband irgendwann während der beiden Studien SOLAR I und SOLAR II gearbeitet hat, unabhängig davon, wie viele Wochenstunden gearbeitet wurde.

Verwendete Variablen:

Variablenname	Frage & Kodierung
GEARB_s1	Haben Sie schon einmal irgendeine Arbeit / irgendeinen Ferienjob gehabt? 0=Nein, 1=Ja, NA=Missing
GEARB_s2	Haben Sie seit der letzten SOLAR-Studie (2003/2004) irgendeine Arbeit/irgendeinen Ferienjob für mindestens 1 Monat gehabt? 0=Nein, 1=Ja, NA=Missing

Bildung der Variable GEARB_s1s2 (pro Proband)

Kodierungsbeschreibung:
Wurde in keiner der beiden Studien gearbeitet, dann wird die Variable GEARB_s1s2 auf 0 gesetzt. Wurde in mind. einer Studie gearbeitet, dann wird die Variable GEARB_s1s2 auf 1 gesetzt. Fehlt die Angabe, ob gearbeitet wurde zu einer Studie und während der anderen Studie wurde nicht gearbeitet, so wird der Wert auf missing (NA) gesetzt. Fehlen die Angaben zu beiden Studien, so wird der Wert ebenfalls auf missing (NA) gesetzt.

Kodierungsübersicht:

Verwendete Variablen		Abgeleitete Variablen
GEARB_s1	**GEARB_s2**	**GEARB_s1s2**
0	0	0
0,1,NA	1	1
1	0,1,NA	1
0	NA	NA
NA	0	NA
NA	NA	NA

Code-Übersicht:

Variablenname	Frage & Kodierung
GEARB_s1s2	gearbeitet während SOLAR I und SOLAR II (unabhängig von der Anzahl der Wochenstunden) 0=Nein, 1=Ja, NA=Missing

R-Code:

```
> GEARB_s1s2 <- NA
> GEARB_s1s2[!is.na(GEARB_s1) & !is.na(GEARB_s2)
+ & (GEARB_s1==0) & (GEARB_s2==0)] <- 0
> GEARB_s1s2[!is.na(GEARB_s1) & !is.na(GEARB_s2)
+ & (GEARB_s1==0) & (GEARB_s2==1)] <- 1
> GEARB_s1s2[!is.na(GEARB_s1) & !is.na(GEARB_s2)
+ & (GEARB_s1==1) & (GEARB_s2==0)] <- 1
> GEARB_s1s2[!is.na(GEARB_s1) & !is.na(GEARB_s2)
+ & (GEARB_s1==1) & (GEARB_s2==1)]<-1
> GEARB_s1s2[is.na(GEARB_s1) & !is.na(GEARB_s2)
+ & (GEARB_s2==1)] <- 1
> GEARB_s1s2[!is.na(GEARB_s1) & is.na(GEARB_s2)
+ & (GEARB_s1==1)] <- 1
```

A.4.4 Ende der Tätigkeit in SOLAR-I

Bei den Datensätzen, bei denen nur die Angaben zum Ende der Tätigkeit fehlte und die restlichen Tätigkeitsangaben vollständig waren (d.h. ISCO-Code, Anfang der Tätigkeit und Wochenstunden) wird davon ausgegangen, dass diese Person die Tätigkeit zum Zeitpunkt der Befragung noch ausführt. Aus diesem Grund wird hierfür ein Ersatzende eingesetzt. Als Ersatzende wurde falls Vorhanden das Ausfülldatum, ansonsten das Einscanndatum des Fragebogens verwendet.

Bildung der Variablen END_MONATx und END_JAHRx (pro Tätigkeit)

Kodierungsbeschreibung:
In die neue Variable END_MONATx bzw. END_JAHRx wurde zunächst das tatsächlich angegebenen Ende der Tätigkeit eingefügt (END_MONAT und END_JAHR). Fehlte diese Angabe, so wurde das Ersatzende verwendet (s1Ersatzende_Monat und s1Ersatzende_Jahr).

Kodierungsübersicht:

Verwendete Variablen		**Abgeleitete Variablen**
END_MONAT **END_JAHR**	**s1Ersatzende_Monat** **s1Ersatzende_Jahr**	**END_MONATx** **END_JAHRx**
Angaben	Ersatzende	Angaben
NA	Ersatzende	Ersatzende
NA	NA	NA

R-Code:

```
> END_MONATx <- END_MONAT
> END_JAHRx <- END_JAHR
> for (i in 1:nrow()){
+ if((ISCO[i]!="9999") & !is.na(ANF_MONAT[i]) & !is.na(ANF_JAHR[i])
+ & is.na(END_MONAT[i]) & is.na(END_JAHR[i])  & !is.na(WST[i])){
+ END_MONATx[i] <- s1Ersatzende_Monat[i]
+ END_JAHRx[i] <- s1Ersatzende_Jahr[i]
+ }
+ }
```

A.4.5 Ende der Tätigkeit in SOLAR II

Bei den Datensätzen, bei denen nur die Angaben zum Ende der Tätigkeit fehlte und die restlichen Tätigkeitsangaben vollständig waren (d.h. ISCO-Code, Anfang der Tätigkeit und Wochenstunden) wird davon ausgegangen, dass diese Person die Tätigkeit zum Zeitpunkt der Befragung noch ausführt. Aus diesem Grund wird hierfür ein Ersatzende eingesetzt. Als Ersatzende wurde falls Vorhanden das Ausfülldatum, ansonsten das Einscanndatum des Fragebogens verwendet.

Bildung der Variablen END_MONATx und END_JAHRx (pro Tätigkeit)

Kodierungsbeschreibung:
In die neue Variable END_MONATx bzw. END_JAHRx wurde zunächst das tatsächlich angegebenen Ende der Tätigkeit (Ende_Monat und Ende_Jahr) eingefügt. Fehlte diese Angabe, so wurde das Ersatzende verwendet (s2Ersatzende_Monat und s2Ersatzende_Jahr).

Kodierungsübersicht:

Verwendete Variablen		**Abgeleitete Variablen**
Ende_Monat **Ende_Jahr**	**s2Ersatzende_Monat** **s2Ersatzende_Jahr**	**END_MONATx** **END_JAHRx**
Angaben	Ersatzende	Angaben
NA	Ersatzende	Ersatzende
NA	NA	NA

R-Code:

```
> END_MONATx <- Ende_Monat
> END_JAHRx <- Ende_Jahr
> for (i in 1:nrow()){
+ if((ISCO[i]!="9999") & (ISCO[i]!="8888") & !is.na(Beginn_Monat[i])
+ & !is.na(Beginn_Jahr[i]) & is.na(Ende_Monat[i]) & is.na(Ende_Jahr[i])
+ & !is.na(Wochenstunden[i])){
+ END_MONATx[i] <- s2Ersatzende_Monat[i]
+ END_JAHRx[i] <- s2Ersatzende_Jahr[i]
+ }
+ }
```

A.4.6 Jemals (mind. acht Wochenstunden) gearbeitet in SOLAR I und SOLAR II

In drei Schritten, die im Folgenden ausführlich dargestellt werden, wurde aus den Tätigkeitsangaben aus SOLAR I und SOLAR II die Variable JEMALS_GEARB gebildet. Jemand, der während SOLAR I und SOLAR II mind. eine Tätigkeit mit mind. acht Wochenstunden durchgeführt hat, wird hier mit "Ja" kodiert.

Gebildete Variable:

Variablenname	Frage & Kodierung
JEMALS_GEARB	Jemals (mind. acht Wochenstunden) gearbeitet in SOLAR-I und SOLAR-II 0=Nein, 1=Ja

Schritt 1: Bildung der Variable MIND_8WST (pro Tätigkeit)

Kodierungsbeschreibung:
Betragen die angegebenen Wochenstunden der jeweiligen Tätigkeit mindestens acht Stunden, so wird die Variable MIND_8WST für diese Tätigkeit auf 1 gesetzt. Ansonsten wird die Variable auf 0 gesetzt. Fehlt die Angabe zu den Wochenstunden, so wird auch die Variable MIND_8WST auf NA gesetzt.

Kodierungsübersicht:

Verwendete Variablen	Abgeleitete Variablen
WST	**MIND_8WST**
< 8	0
≥ 8	1
NA	NA

Code-Übersicht:

Variablenname	Frage & Kodierung
MIND_8WST	jeweilige Tätigkeit mind. 8 Wochenstunden ausgeführt 0=Nein, 1=Ja, NA=Missing

R-Code:

```
> MIND_8WST <- 0
> MIND_8WST[!is.na(WST)&WST>=8]<-1
> MIND_8WST[is.na(WST)]<-NA
```

Schritt 2: Bildung der Variable SUM_BERUF_MIND_8WST (pro Proband)

Kodierungsbeschreibung:
Alle Tätigkeiten eines Probanden, die mind. acht Wochenstunden durchgeführt wurden, werden aufsummiert und in der Variable SUM_BERUF_MIND_8WST abgespeichert.

Code-Übersicht:

Variablenname	Frage & Kodierung
SUM_BERUF_MIND_8WST	Summe der Tätigkeiten, die mind. acht Wochenstunden ausgeführt wurden 0-10=Summe

R-Code:

```
> j <- 1
> i <- 1
> while (j <= nrow()) {
+ SUMME <- sum(MIND_8WST[j],MIND_8WST[j+1],MIND_8WST[j+2],MIND_8WST[j+3],MIND_8WST[j+4],MIND_8WST[j+5],
+ MIND_8WST[j+6],MIND_8WST[j+7],MIND_8WST[j+8],MIND_8WST[j+9],na.rm=TRUE)
+ SUM_BERUF_MIND_8WST[i] <- SUMME
+ SUM_BERUF_MIND_8WST[i+1] <- SUMME
+ SUM_BERUF_MIND_8WST[i+2] <- SUMME
+ SUM_BERUF_MIND_8WST[i+3] <- SUMME
+ SUM_BERUF_MIND_8WST[i+4] <- SUMME
+ SUM_BERUF_MIND_8WST[i+5] <- SUMME
+ SUM_BERUF_MIND_8WST[i+6] <- SUMME
+ SUM_BERUF_MIND_8WST[i+7] <- SUMME
+ SUM_BERUF_MIND_8WST[i+8] <- SUMME
+ SUM_BERUF_MIND_8WST[i+9] <- SUMME
+ j <- j+10
+ i <- i+10
+ }
```

Schritt 3: Bildung der Variable JEMALS_GEARB (pro Proband)

Kodierungsbeschreibung:
Hat ein Proband mind. eine Tätigkeit mit mind. acht Wochenstunden während der Studien SOLAR I oder SOLAR II durchgeführt, so wird die Variable JEMALS_GEARB für denjenigen Probanden auf 1 gesetzt. Ansonsten wird sie auf 0 gesetzt.

Kodierungsübersicht:

Verwendete Variablen	Abgeleitete Variablen
SUM_BERUF_MIN_8WST	**JEMALS_GEARB**
0	0
≥ 0	1

Code-Übersicht:

Variablenname	Frage & Kodierung
JEMALS_GEARB	Jemals (mind. acht Wochenstunden) gearbeitet in SOLAR I und SOLAR II 0=Nein, 1=Ja

R-Code:

```
> JEMALS_GEARB <- 0
> JEMALS_GEARB[SUM_BERUF_MIND_8WST>0] <- 1
```

A.4.7 Anzahl Tätigkeitsangaben in SOLAR I und SOLAR II

In mehreren Schritten, die im Folgenden ausführlich dargestellt werden, wurde aus den Tätigkeitsangaben aus SOLAR I und SOLAR II die Variable SUM_ANZAHL_EINTRAEGE gebildet, die angibt, wie viele Tätigkeiten der Proband genannt hat (unabhängig davon, wie viele Wochenstunden gearbeitet wurden).

Gebildete Variable:

Variablenname	Frage & Kodierung
SUM_ANZAHL_EINTRAEGE	Anzahl der Tätigkeiten (unabhängig von der Wochenstundenanzahl) 0-10=Summe

Schritt 1: Bildung der Variable ANZAHL_EINTRAEGE (pro Tätigkeit) in SOLAR I

Kodierungsbeschreibung:
Die Variable ANZAHL_EINTRAEGE gibt wieder, ob ein Proband in einer Zeile Tätigkeitsangaben gemacht hat (=1) oder nicht (=0).

Kodierungsübersicht:

Verwendete Variablen	Abgeleitete Variablen
ISCO	**ANZAHL_EINTRAEGE**
9999	0
alle anderen Codes	1

Code-Übersicht:

Variablenname	Frage & Kodierung
ANZAHL_EINTRAEGE	Tätigkeit angegeben 0=Nein, 1=Ja

R-Code:

```
> ANZAHL_EINTRAEGE <- 0
> ANZAHL_EINTRAEGE[ISCO != 9999] <- 1
```

Schritt 2: Bildung der Variable ANZAHL_EINTRAEGE (pro Tätigkeit) in SOLAR II

Kodierungsbeschreibung:
Die Variable ANZAHL_EINTRAEGE gibt wieder, ob ein Proband in einer Zeile Tätigkeitsangaben gemacht hat (=1) oder nicht (=0).

Kodierungsübersicht:

Verwendete Variablen	Abgeleitete Variablen
ISCO	**ANZAHL_EINTRAEGE**
8888	0
9999	0
alle anderen Codes	1

Zusätzlich wurde die Variable ANZAHL_EINTRAEGE auf 1 gesetzt, wenn die Antwortoption “Keine Tätigkeit für mind. 8 Stunden pro Woche ausgeführt” angegeben wurde (s2f93_01 = 1).

Code-Übersicht:

Variablenname	Frage & Kodierung
ANZAHL_EINTRAEGE	Tätigkeit angegeben 0=Nein, 1=Ja

R-Code:

```
> ANZAHL_EINTRAEGE <- 0
> ANZAHL_EINTRAEGE[ISCO != 9999 & ISCO != 8888] <- 1
> ANZAHL_EINTRAEGE[(s2f93_01==1)] <- 1
```

Schritt 3: Bildung der Variable SUM_ANZAHL_EINTRAEGE_s1 (pro Proband) für SOLAR I

Kodierungsbeschreibung:
Alle Tätigkeiten eines Probanden innerhalb von SOLAR I (unabhängig von der Anzahl der Wochenstunden) werden aufsummiert und in der Variable
SUM_ANZAHL_EINTRAEGE_s1 abgespeichert.

Code-Übersicht:

Variablenname	Frage & Kodierung
SUM_ANZAHL_EINTRAEGE_s1	Anzahl der Tätigkeiten in SOLAR-I (unabhängig von der Wochenstundenanzahl) 0-5=Summe

R-Code:

```
> j <- 1
> i <- 1
> while (j <= nrow()) {
+ SUMME <- sum(ANZAHL_EINTRAEGE[j],ANZAHL_EINTRAEGE[j+1],ANZAHL_EINTRAEGE[j+2],ANZAHL_EINTRAEGE[j+3],
+ ANZAHL_EINTRAEGE[j+4], na.rm=TRUE)
+ SUM_ANZAHL_EINTRAEGE_s1[i] <- SUMME
+ SUM_ANZAHL_EINTRAEGE_s1[i+1] <- SUMME
+ SUM_ANZAHL_EINTRAEGE_s1[i+2] <- SUMME
+ SUM_ANZAHL_EINTRAEGE_s1[i+3] <- SUMME
+ SUM_ANZAHL_EINTRAEGE_s1[i+4] <- SUMME
+ j <- j+5
+ i <- i+5
+ }
```

Schritt 4: Bildung der Variable SUM_ANZAHL_EINTRAEGE_s2 (pro Proband) für SOLAR II

Kodierungsbeschreibung:
Alle Tätigkeiten eines Probanden innerhalb von SOLAR II (unabhängig von der Anzahl der Wochenstunden) werden aufsummiert und in der Variable
SUM_ANZAHL_EINTRAEGE_s2 abgespeichert.

Code-Übersicht:

Variablenname	Frage & Kodierung
SUM_ANZAHL_EINTRAEGE_s2	Anzahl der Tätigkeiten in SOLAR II (unabhängig von der Wochenstundenanzahl) 0-5=Summe

R-Code:

```
> j <- 1
> i <- 1
> while (j <= nrow()) {
+ SUMME <- sum(ANZAHL_EINTRAEGE[j],ANZAHL_EINTRAEGE[j+1],ANZAHL_EINTRAEGE[j+2],ANZAHL_EINTRAEGE[j+3],
+ ANZAHL_EINTRAEGE[j+4], na.rm=TRUE)
+ SUM_ANZAHL_EINTRAEGE_s2[i] <- SUMME
+ SUM_ANZAHL_EINTRAEGE_s2[i+1] <- SUMME
+ SUM_ANZAHL_EINTRAEGE_s2[i+2] <- SUMME
+ SUM_ANZAHL_EINTRAEGE_s2[i+3] <- SUMME
+ SUM_ANZAHL_EINTRAEGE_s2[i+4] <- SUMME
+ j <- j+5
+ i <- i+5
+ }
```

Schritt 5: Bildung der Variable SUM_ANZAHL_EINTRAEGE (pro Proband) für SOLAR I und SOLAR II

Kodierungsbeschreibung:
Die Anzahl der Tätigkeiten aus SOLAR I (SUM_ANZAHL_EINTRAEGE_s1) und SOLAR II (SUM_ANZAHL_EINTRAEGE_s2) werden nun aufsummiert und in der Variable
SUM_ANZAHL_EINTRAEGE abgespeichert.

Code-Übersicht:

Variablenname	**Frage & Kodierung**
SUM_ANZAHL_EINTRAEGE	Anzahl der Tätigkeiten (unabhängig von der Wochenstundenanzahl) 0-10=Summe

R-Code:

```
> SUM_ANZAHL_EINTRAEGE <- SUM_ANZAHL_EINTRAEGE_s1 + SUM_ANZAHL_EINTRAEGE_s2
```

A.4.8 Dauer der Tätigkeit

Die Dauer der jeweiligen Tätigkeit wurde berechnet, indem jeweils das Ende der Tätigkeit vom Anfang der Tätigkeit abgezogen wurde. Die genaue Berechnung musste in einer längeren Schleife programmiert werden und ist deshalb nicht hier dargestellt, sondern kann dem R-Code (auf der beigelegten CD) entnommen werden.

A.4.9 Zeilen mit vollständig ausgefüllten Tätigkeitsangaben

In mehreren Schritten, die im Folgenden ausführlich dargestellt werden, wurde aus den Tätigkeitsangaben aus SOLAR I und SOLAR II die Variable SUM_ZEILE_VOLLST gebildet, die angibt, wie viele Zeilen mit Tätigkeitsangaben der Proband vollständig ausgefüllt hat. Eine Zeile gilt als vollständig ausgefüllt, wenn der ISCO-Code 94, 95, 97 oder 98 auftritt, nie gearbeitet wurde oder wenn alle Jobs, die angegeben wurden, vollständig ausgefüllt sind (d.h. Angaben zu Beginn und Ende der Tätigkeit, Wochenstunden und der ISCO-Code vorliegen). Weiterhin gilt die Zeile als vollständig, wenn weniger als acht Wochenstunden gearbeitet wurde oder die Antwortoption "Keine Tätigkeit für mind. 8 Stunden pro Woche ausgeführt" ausgewählt wurde.

Gebildete Variable:

Variablenname	Frage & Kodierung
SUM_ZEILE_VOLLST	Summe der vollständig ausgefüllten Zeilen 0-10=Summe

Schritt 1: Bildung der Variable ZEILE_VOLLST (pro Tätigkeit) in SOLAR I

Kodierungsbeschreibung:
Zunächst wird eine Hilfsvariable ZEILE_VOLLST für SOLAR I gebildet, die angibt, ob die Zeile vollständig ausgefüllt wurde (=1) oder später imputiert werden muss (=0).

Code-Übersicht:

Variablenname	Frage & Kodierung
ZEILE_VOLLST	Zeile vollständig ausgefüllt 0=Nein, 1=Ja

R-Code:

```
> ZEILE_VOLLST <- 0
> ZEILE_VOLLST[!is.na(ANF_JAHR) & !is.na(ANF_MONAT) & !is.na(END_JAHRx)
+ & !is.na(END_MONATx) & !is.na(WST) & !is.na(ISCO)] <- 1
> ZEILE_VOLLST[ISCO==94 | ISCO==95 | ISCO==98 | ISCO==97] <- 1
> ZEILE_VOLLST[WST<8] <- 1
```

Schritt 2: Bildung der Variable ZEILE_VOLLST (pro Tätigkeit) in SOLAR II

Kodierungsbeschreibung:
Zunächst wird eine Hilfsvariable ZEILE_VOLLST für SOLAR II gebildet, die angibt, ob die Zeile vollständig ausgefüllt wurde (=1) oder später imputiert werden muss (=0).

Code-Übersicht:

Variablenname	Frage & Kodierung
ZEILE_VOLLST	Zeile vollständig ausgefüllt 0=Nein, 1=Ja

R-Code:

```
> ZEILE_VOLLST <- 0
> ZEILE_VOLLST[!is.na(ANF_JAHR) & !is.na(ANF_MONAT) & !is.na($END_JAHRx)
+ & !is.na(END_MONATx) & !is.na(WST) & !is.na(ISCO)] <- 1
> ZEILE_VOLLST[$ISCO==94 | ISCO==95 | ISCO==98 | ISCO==97] <- 1
> ZEILE_VOLLST[WST<8] <- 1
> ZEILE_VOLLST[(s2f93_01==1)] <- 1
```

Schritt 3: Bildung der Variable SUM_ZEILE_VOLLST_s1 (pro Proband) für SOLAR I

Kodierungsbeschreibung:
Aus der Hilfsvariable ZEILE_VOLLST wird dann für SOLAR I eine Variable
SUM_ZEILE_VOLLST_s1 gebildet, die pro Probanden alle fünf möglichen Zeilen aufsummiert und die Summe der vollständig ausgefüllten Zeilen enthält.

Code-Übersicht:

Variablenname	Frage & Kodierung
SUM_ZEILE_VOLLST_s1	Summe der vollständig ausgefüllten Zeilen in SOLAR I 0-5=Summe

R-Code:

```
> j <- 1
> i <- 1
> while (j <= nrow()) {
+ SUMME <- sum(ZEILE_VOLLST[j],ZEILE_VOLLST[j+1],ZEILE_VOLLST[j+2],
+ ZEILE_VOLLST[j+3],ZEILE_VOLLST[j+4],na.rm=TRUE)
+ SUM_ZEILE_VOLLST_s1[i] <- SUMME
+ SUM_ZEILE_VOLLST_s1[i+1] <- SUMME
+ SUM_ZEILE_VOLLST_s1[i+2] <- SUMME
+ SUM_ZEILE_VOLLST_s1[i+3] <- SUMME
+ SUM_ZEILE_VOLLST_s1[i+4] <- SUMME
+ j <- j+5
+ i <- i+5
+ }
```

Schritt 4: Bildung der Variable SUM_ZEILE_VOLLST_s2 (pro Proband) für SOLAR II

Kodierungsbeschreibung:
Aus der Hilfsvariable ZEILE_VOLLST wird dann für SOLAR II eine Variable
SUM_ZEILE_VOLLST_s2 gebildet, die pro Probanden alle fünf möglichen Zeilen aufsummiert und die Summe der vollständig ausgefüllten Zeilen enthält.

Code-Übersicht:

Variablenname	Frage & Kodierung
SUM_ZEILE_VOLLST_s2	Summe der vollständig ausgefüllten Zeilen in SOLAR II 0-5=Summe

R-Code:

```
> j <- 1
> i <- 1
> while (j <= nrow()) {
+ SUMME <- sum(ZEILE_VOLLST[j],ZEILE_VOLLST[j+1],ZEILE_VOLLST[j+2],
+ ZEILE_VOLLST[j+3],ZEILE_VOLLST[j+4],na.rm=TRUE)
+ SUM_ZEILE_VOLLST_s2[i] <- SUMME
+ SUM_ZEILE_VOLLST_s2[i+1] <- SUMME
+ SUM_ZEILE_VOLLST_s2[i+2] <- SUMME
+ SUM_ZEILE_VOLLST_s2[i+3] <- SUMME
+ SUM_ZEILE_VOLLST_s2[i+4] <- SUMME
+ j <- j+5
+ i <- i+5
+ }
```

Schritt 5: Bildung der Variable SUM_ZEILE_VOLLST (pro Proband) für SOLAR I und SOLAR II

Kodierungsbeschreibung:
Aus den Variablen SUM_ZEILE_VOLLST_s1 und SUM_ZEILE_VOLLST_s2 wird dann eine gemeinsame Variable SUM_ZEILE_VOLLST gebildet, die pro Probanden alle zehn möglichen Zeilen aus SOLAR I und SOLAR II aufsummiert und die Summe der vollständig ausgefüllten Zeilen enthält.

Code-Übersicht:

Variablenname	**Frage & Kodierung**
SUM_ZEILE_VOLLST	Summe der vollständig ausgefüllten Zeilen 0-10=Summe

R-Code:

```
> SUM_ZEILE_VOLLST <- SUM_ZEILE_VOLLST_s1 + SUM_ZEILE_VOLLST_s2
```

A.4.10 Probanden mit vollständig ausgefüllten Tätigkeitsangaben

In mehreren Schritten, die im Folgenden ausführlich dargestellt werden, wurde aus den Tätigkeitsangaben aus SOLAR I und SOLAR II die Variable PROB_VOLLST gebildet, die angibt, ob ein Proband alle gemachten Tätigkeitsdaten vollständig ausgefüllt hat. Die Probanden die alle gemachten Tätigkeitsdaten vollständig ausgefüllt haben, erhalten in der Variable PROB_VOLLST den Eintrag 1. Fehlt mind. eine Angabe zu den Tätigkeitsdaten, so erhält der Proband den Eintrag 0. Bei all diesen Probanden mit Eintrag 0 können in einem späteren Schritt die fehlenden Angaben imputiert werden.

Gebildete Variable:

Variablenname	Frage & Kodierung
PROB_VOLLST	Proband mit vollständig ausgefüllten Tätigkeitsangaben 0=Nein, 1=Ja

Schritt 1: Bildung der Variable PROB_VOLLST_s1 (pro Proband) in SOLAR I

Kodierungsbeschreibung:
Zunächst wird eine Hilfsvariable PROB_VOLLST_s1 für SOLAR I gebildet, die angibt, ob der Proband in SOLAR I alle gemachten Tätigkeitsangaben vollständig ausgefüllt hat. Ein Proband gilt als vollständig, wenn er in SOLAR-I nicht gearbeitet hat und auch keine Tätigkeitsangaben gemacht hat. Wurde in SOLAR I gearbeitet, so muss die Anzahl der ausgefüllten Zeilen und die Anzahl der vollständig ausgefüllten Zeilen bei diesen Probanden übereinstimmen (SUM_ANZAHL_EINTRAEGE = SUM_ZEILE_VOLLST). Ist das der Fall, gelten sie als vollständig. Weiterhin gelten die Probanden als vollständig, wenn zwar gearbeitet wurde, aber keine Einträge vorliegen oder wenn die Angabe, ob gearbeitet wurde komplett fehlt. In diesen Fällen wird konservativ vorgegangen und somit keine Tätigkeiten und mögliche Expositionen imputiert. Als unvollständig gilt ein Proband in SOLAR I, wenn die gemachten Tätigkeitsangaben fehlende Werte aufweisen.

Code-Übersicht:

Variablenname	Frage & Kodierung
PROB_VOLLST_s1	Proband vollständig in SOLAR I 0=Nein, 1=Ja

R-Code:

```
> PROB_VOLLST_s1 <- NA
> PROB_VOLLST_s1[GEARB_s1 == 0] <- 1
> PROB_VOLLST_s1[(GEARB_s1==1) & (SUM_ANZAHL_EINTRAEGE_s1!=0)
+ & (SUM_ZEILE_VOLLST_s1!=0)
+ & (SUM_ANZAHL_EINTRAEGE_s1==SUM_ZEILE_VOLLST_s1)] <- 1
> PROB_VOLLST_s1[(GEARB_s1==1)
+ & (SUM_ANZAHL_EINTRAEGE_s1!=SUM_ZEILE_VOLLST_s1)] <- 0
> PROB_VOLLST_s1[(GEARB_s1==1) & (SUM_ANZAHL_EINTRAEGE_s1==0)
+ & (SUM_ZEILE_VOLLST_s1==0)] <- 1
> PROB_VOLLST_s1[is.na(GEARB_s1)] <- 1
```

Schritt 2: Bildung der Variable PROB_VOLLST_s2 (pro Proband) in SOLAR II

Kodierungsbeschreibung:
Zunächst wird eine Hilfsvariable PROB_VOLLST_s2 für SOLAR II gebildet, die angibt, ob der Proband in SOLAR II alle gemachten Tätigkeitsangaben vollständig ausgefüllt hat. Ein Proband gilt als vollständig, wenn er in SOLAR-II nicht gearbeitet hat und auch keine Tätigkeitsangaben gemacht hat. Wurde in SOLAR II gearbeitet, so muss die Anzahl der ausgefüllten Zeilen und die Anzahl der vollständig ausgefüllten Zeilen bei diesen Probanden übereinstimmen (SUM_ANZAHL_EINTRAEGE = SUM_ZEILE_VOLLST). Ist das der Fall, gelten sie als vollständig. Weiterhin gelten die Probanden als vollständig, wenn zwar gearbeitet wurde, aber keine Einträge vorliegen oder wenn die Angabe, ob gearbeitet wurde komplett fehlt. In diesen Fällen wird konservativ vorgegangen und somit keine Tätigkeiten und mögliche Expositionen imputiert. Als unvollständig gilt ein Proband in SOLAR II, wenn die gemachten Tätigkeitsangaben fehlende Werte aufweisen.

Code-Übersicht:

Variablenname	Frage & Kodierung
PROB_VOLLST_s2	Proband vollständig in SOLAR II
	0=Nein, 1=Ja

R-Code:

```
> PROB_VOLLST_s2 <- NA
> PROB_VOLLST_s2[GEARB_s2 == 0] <- 1
> PROB_VOLLST_s2[(GEARB_s2==1) & (SUM_ANZAHL_EINTRAEGE_s2!=0)
+ & (SUM_ZEILE_VOLLST_s2!=0)
+ & (SUM_ANZAHL_EINTRAEGE_s2==SUM_ZEILE_VOLLST_s2)] <- 1
> PROB_VOLLST_s2[(GEARB_s2==1)
+ & (SUM_ANZAHL_EINTRAEGE_s2!=SUM_ZEILE_VOLLST_s2)] <- 0
> PROB_VOLLST_s2[(GEARB_s2==1) & (SUM_ANZAHL_EINTRAEGE_s2==0)
+ & (SUM_ZEILE_VOLLST_s2==0)] <- 1
> PROB_VOLLST_s2[is.na(GEARB_s2)] <- 1
```

Schritt 3: Bildung der Variable PROB_VOLLST (pro Proband) für SOLAR I und SOLAR II

Kodierungsbeschreibung:
Aus den Variablen PROB_VOLLST_s1 und PROB_VOLLST_s2 wird eine Variable
PROB_VOLLST gebildet, die angibt, ob der Proband sowohl in SOLAR I als auch in SOLAR II vollständig ist.

Code-Übersicht:

Variablenname	Frage & Kodierung
PROB_VOLLST	Proband in SOLAR I und SOLAR II vollständig
	0=Nein, 1=Ja

R-Code:

```
> PROB_VOLLST <- NA
> PROB_VOLLST[PROB_VOLLST_s1 == 1 & PROB_VOLLST_s2 == 1] <- 1
> PROB_VOLLST[PROB_VOLLST_s1 != 1 | PROB_VOLLST_s2 != 1] <- 0
```

A.5 Benötigte Variable für die Simulation

Für die Simulation wurde eine Variable benötigt, die angibt, ob in der entsprechenden Zeile im Datensatz, der die Probanden mit vollständigen Tätigkeitsangaben enthält, Werte künstlich gelöscht wurden oder nicht. In den Zeilen, in denen Werte künstlich gelöscht wurden muss anschließend auch imputiert werden.

Gebildete Variable:

Variablenname	**Frage & Kodierung**
kuenstl_geloescht	Zeile enthält künstlich gelöschte Werte 0=Nein, 1=Ja

A.6 Benötigte Variablen für die Job-Matrix

A.6.1 Kurzbeschreibung der in der Basis-Job-Matrix enthaltenen Variablen

Folgende Variablen sind in der Basis-Job-Matrix enthaltenen:

Variablenname	**Variablenbeschreibung**
knr	Kohortennummer des Probanden
JAHR	Pro Proband und Tätigkeit die Jahre 2000-2009 (bzw. bei den Probanden, die schon vor 2000 gearbeitet haben, die Jahre 1992-2009)
NR_BERUF	Gibt an, in welcher Reihenfolge die Tätigkeiten ausgeübt wurden; Tätigkeit mit NR_BERUF=1 ist die erste Tätigkeit des jeweiligen Probanden
ANF_MONAT	Anfangsmonat der Tätigkeit im entsprechenden Jahr
END_MONATx	Endmonat der Tätigkeit im entsprechenden Jahr
WST	Anzahl der Wochenstunden, die in der entsprechenden Tätigkeit gearbeitet wurden
ISCO	ISCO-Code der Tätigkeit
HMW*	Gibt an, ob Exposition in der Kategorie HMW bestand (0=nein, 1=ja)
GEARB_MON	Gibt an, wie viele Monate im entsprechenden Jahr gearbeitet wurden
MIND_8WST	Gibt an, ob mindestens acht Stunden pro Woche gearbeitet wurde (0=nein: WST $<$ 8, 1=ja: WST $\geq$ 8)
HMW_monat*	Exposition in der Kategorie HMW in Stunden pro Monat: WST * 4,25 * HMW
HMW_jahr*	Exposition in der Kategorie HMW in Stunden pro Jahr: HMW_monat * GEARB_MON

*(Analog für LMW/MIXED/IRRPEAKS/LOWRISK)

A.6.2 Kurzbeschreibung der aus der Basis-Job-Matrix gebildeten Variablen

Auf Grundlage der Basis-Job-Matrix konnten dann die folgende Variablen berechnet werden, die als mögliche Kovariablen in die Regressionsmodelle eingehen können:

Variablenname	**Variablenbeschreibung**
HMW_kumuliert*	kumulierte Exposition pro Proband in der Kategorie HMW über alle Tätigkeiten und Jahre hinweg (in Stunden)
HMW_binaer*	binäre Exposition pro Proband in der Kategorie HMW über alle Tätigkeiten und Jahre hinweg (0=nein, 1=ja)
HMW_erstesjahr_gesamt*	kumulierte Exposition pro Proband in der Kategorie HMW innerhalb des 1. Tätigkeitsjahres (in Stunden)
HMW_erstesjahr_binaer*	binäre Exposition pro Proband in der Kategorie HMW innerhalb des 1. Tätigkeitsjahres (0=nein, 1=ja)
HMW_ersterberuf_gesamt*	kumulierte Exposition pro Proband in der Kategorie HMW während der 1. Tätigkeit (in Stunden)
HMW_ersterberuf_binaer*	binäre Exposition pro Proband in der Kategorie HMW während der 1. Tätigkeit (0=nein, 1=ja)

*(Analog für LMW/MIXED/IRRPEAKS/LOWRISK)

ANHANG B

Alle Abbildungen zum Vergleich der Parameterschätzer

Im Folgenden sind alle Graphiken zum Vergleich der Parameterschätzer in Kapitel 8 getrennt nach den Einflussgrößen des logistischen Modells für die Zielgröße Asthma in SOLAR II abgebildet.

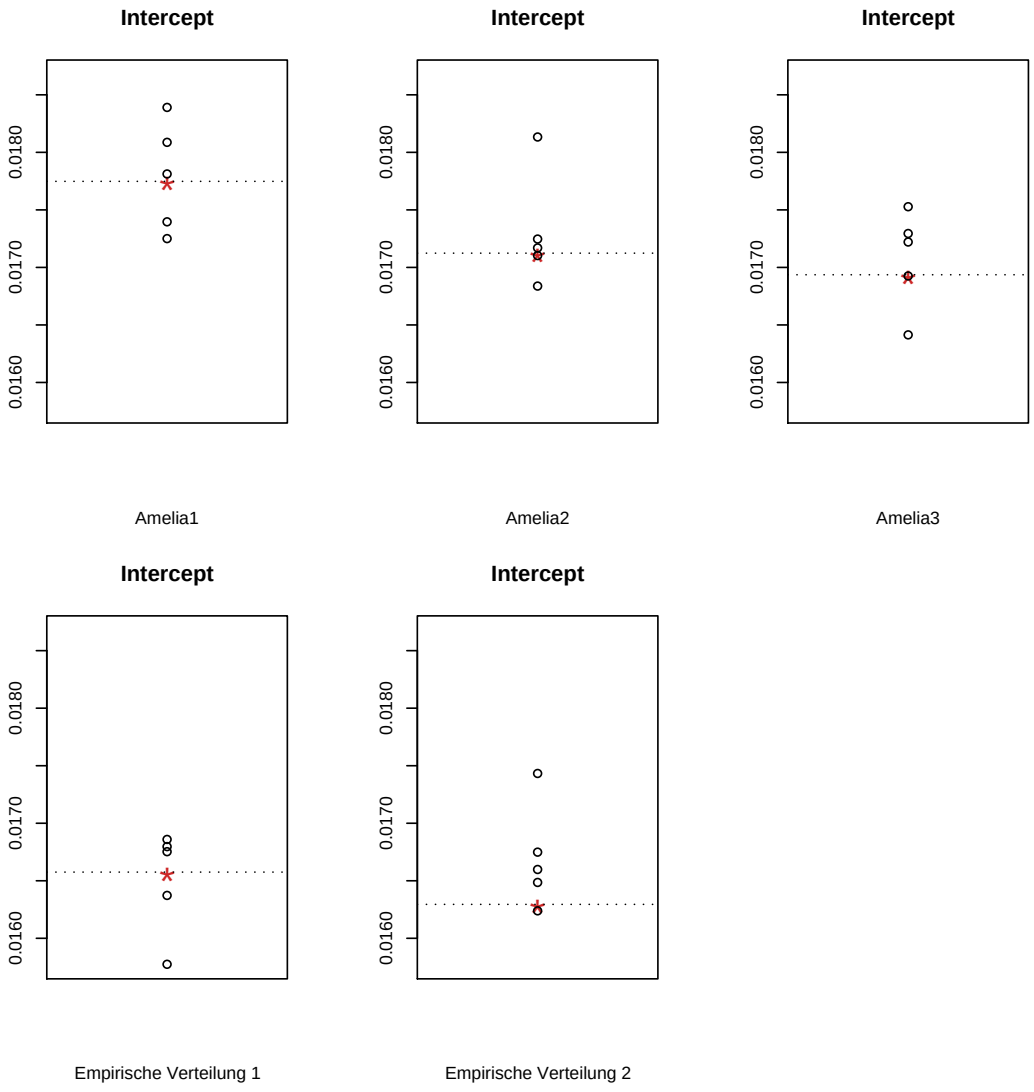


Abbildung B.1: Vergleich der Parameterschätzer - Intercept

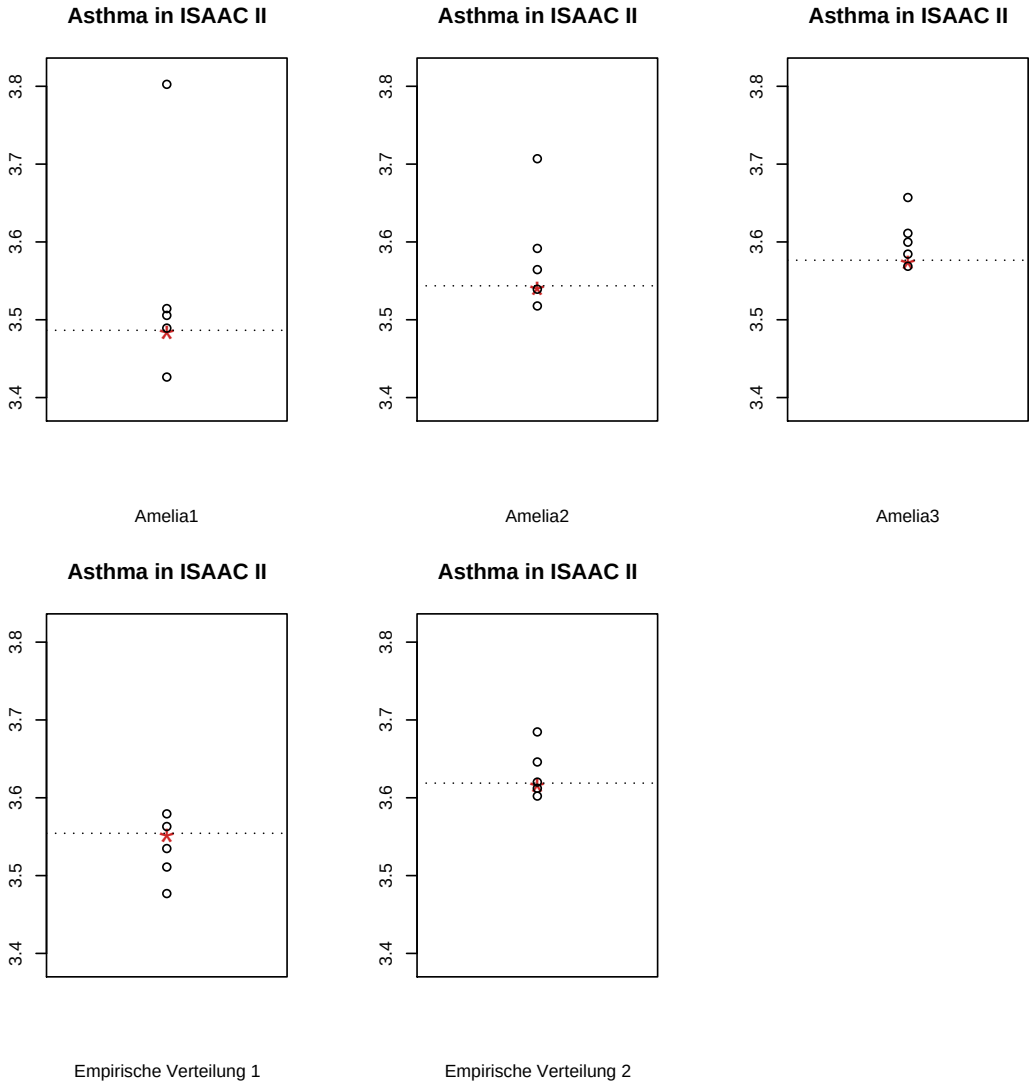


Abbildung B.2: Vergleich der Parameterschätzer - Asthma in ISAAC II

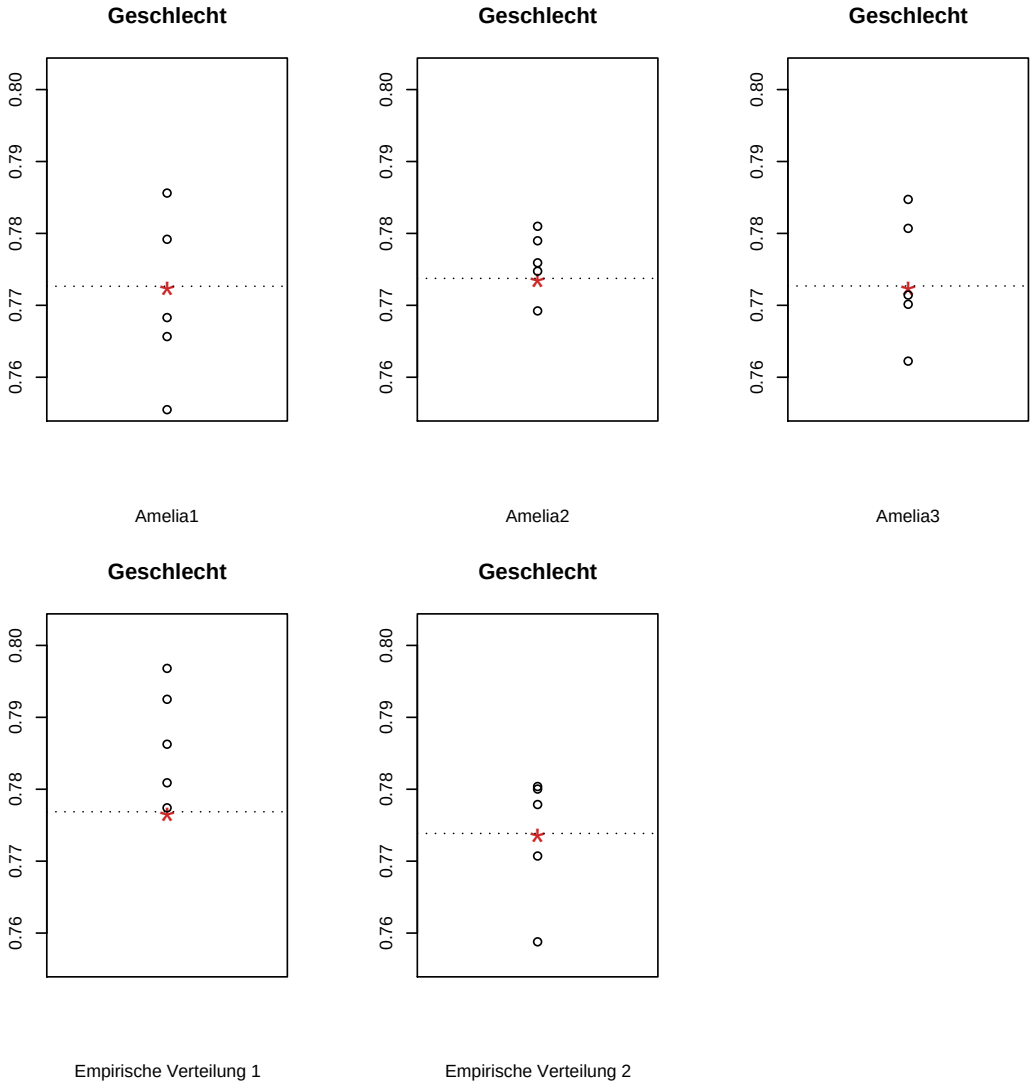


Abbildung B.3: Vergleich der Parameterschätzer - Geschlecht

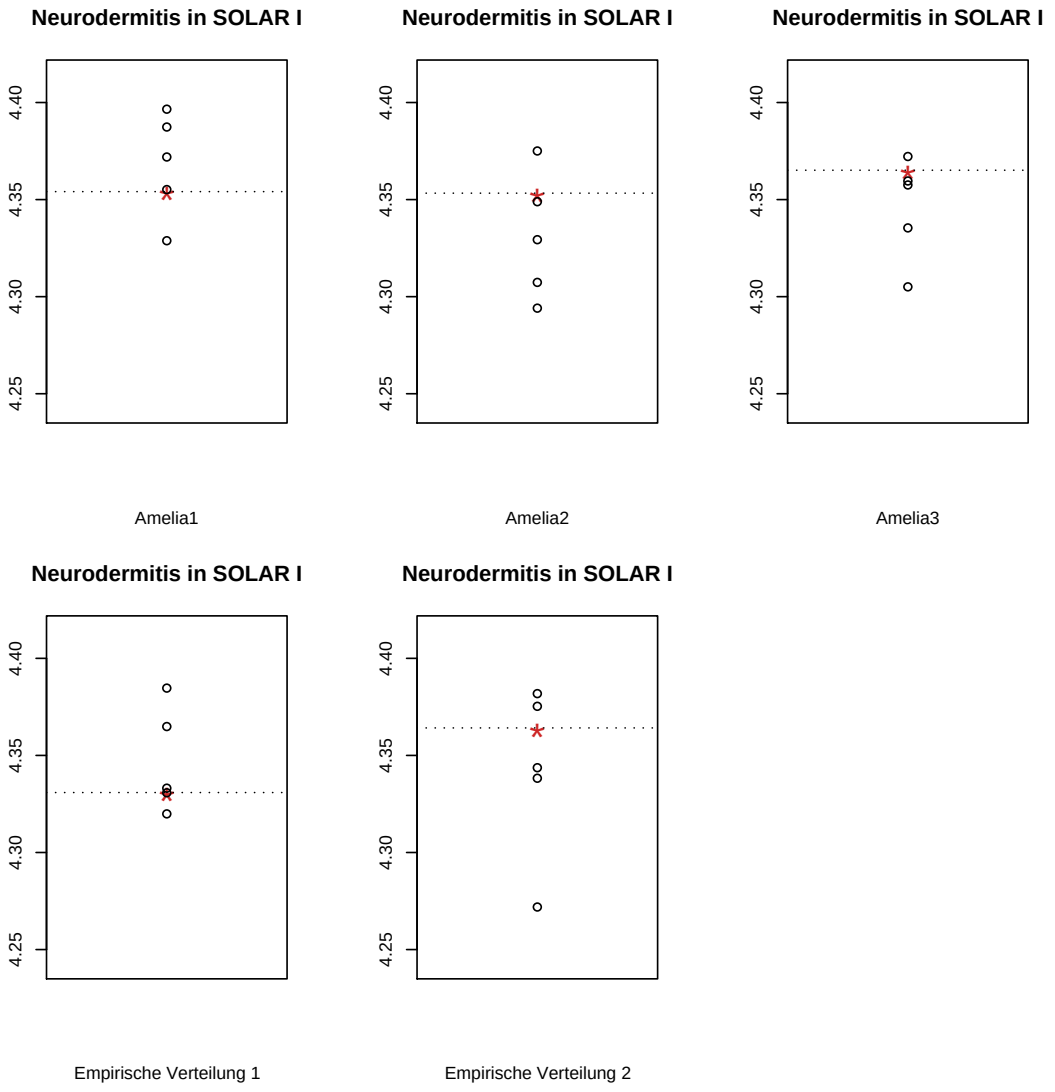


Abbildung B.4: Vergleich der Parameterschätzer - Neurodermitis in SOLAR I

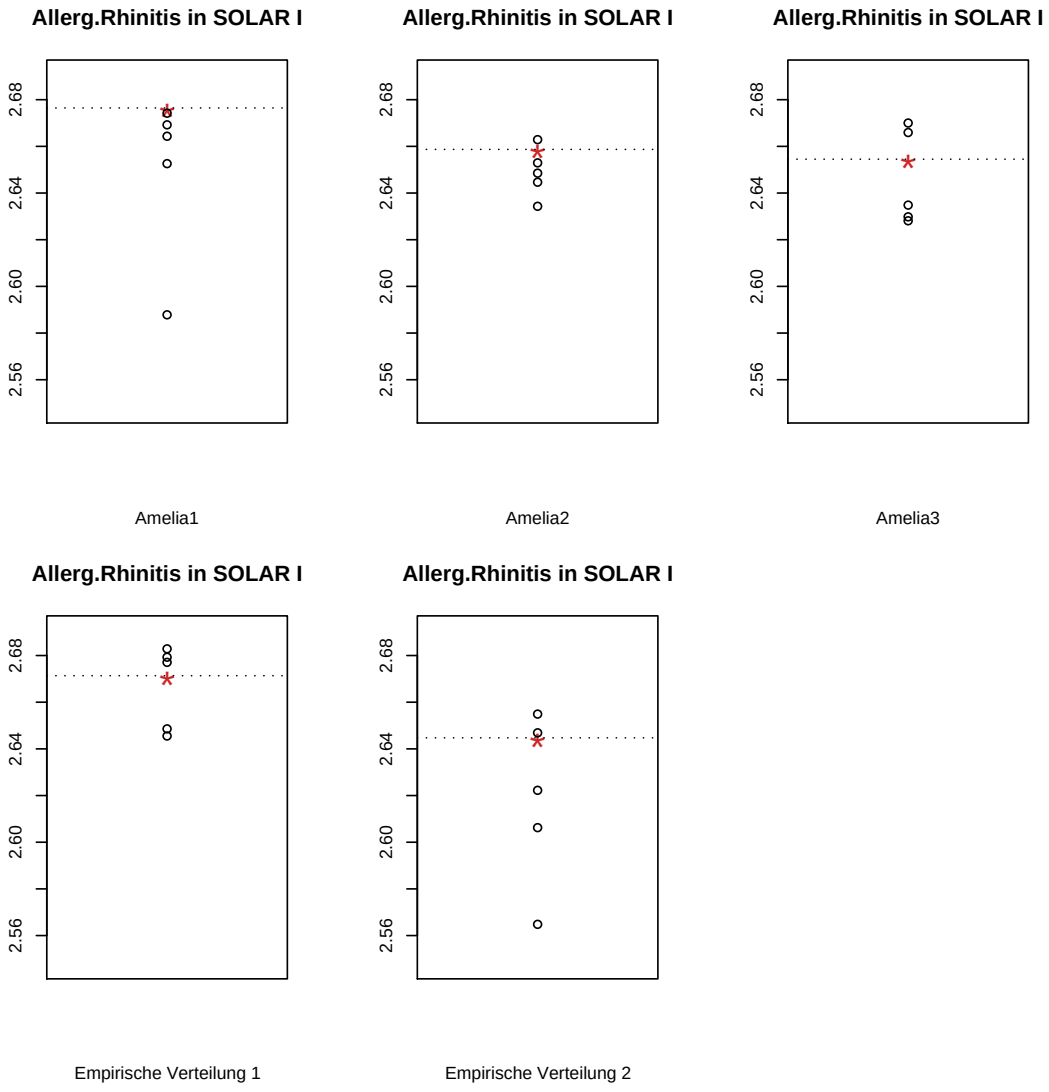


Abbildung B.5: Vergleich der Parameterschätzer - Allergische Rhinitis in SOLAR I

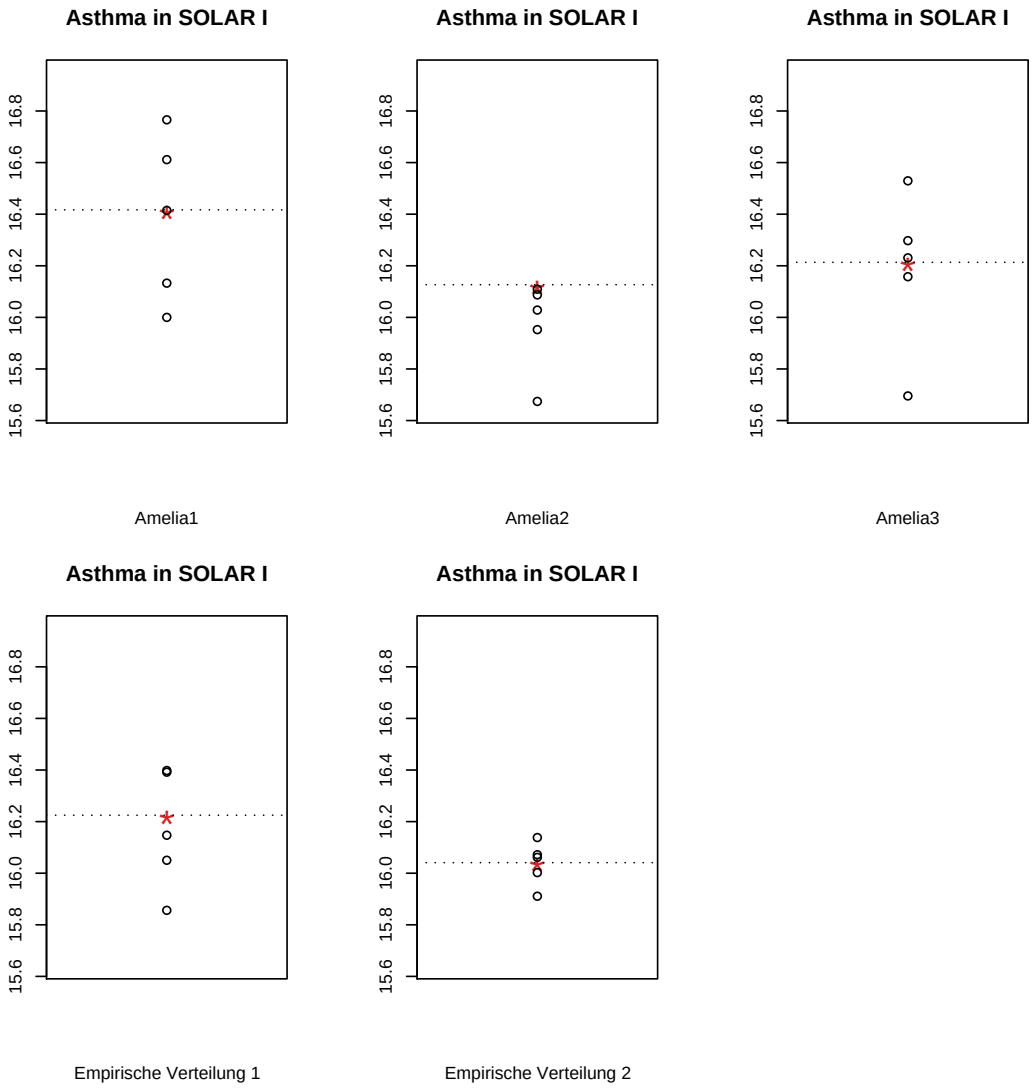


Abbildung B.6: Vergleich der Parameterschätzer - Asthma in SOLAR I

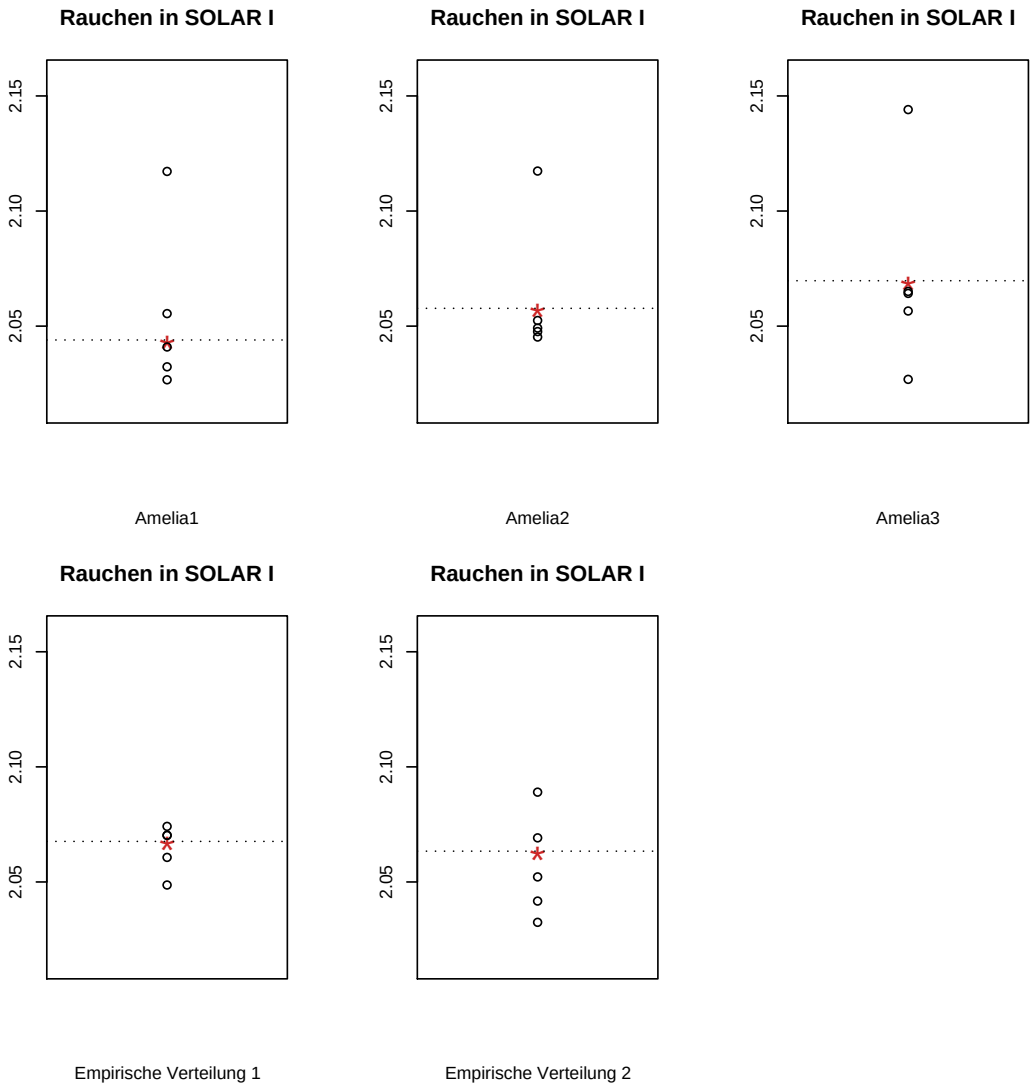


Abbildung B.7: Vergleich der Parameterschätzer - Rauchen in SOLAR I

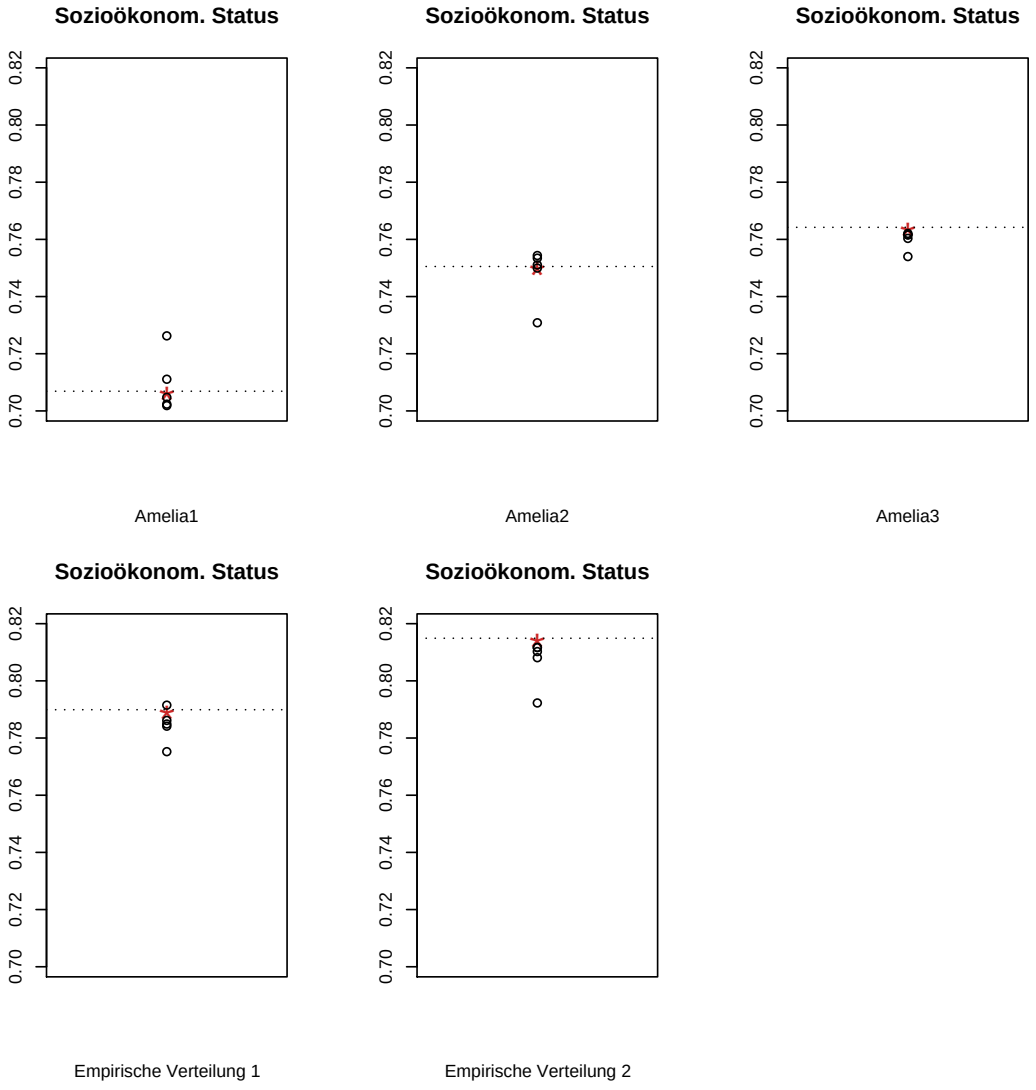


Abbildung B.8: Vergleich der Parameterschätzer - Sozioökonomischer Status

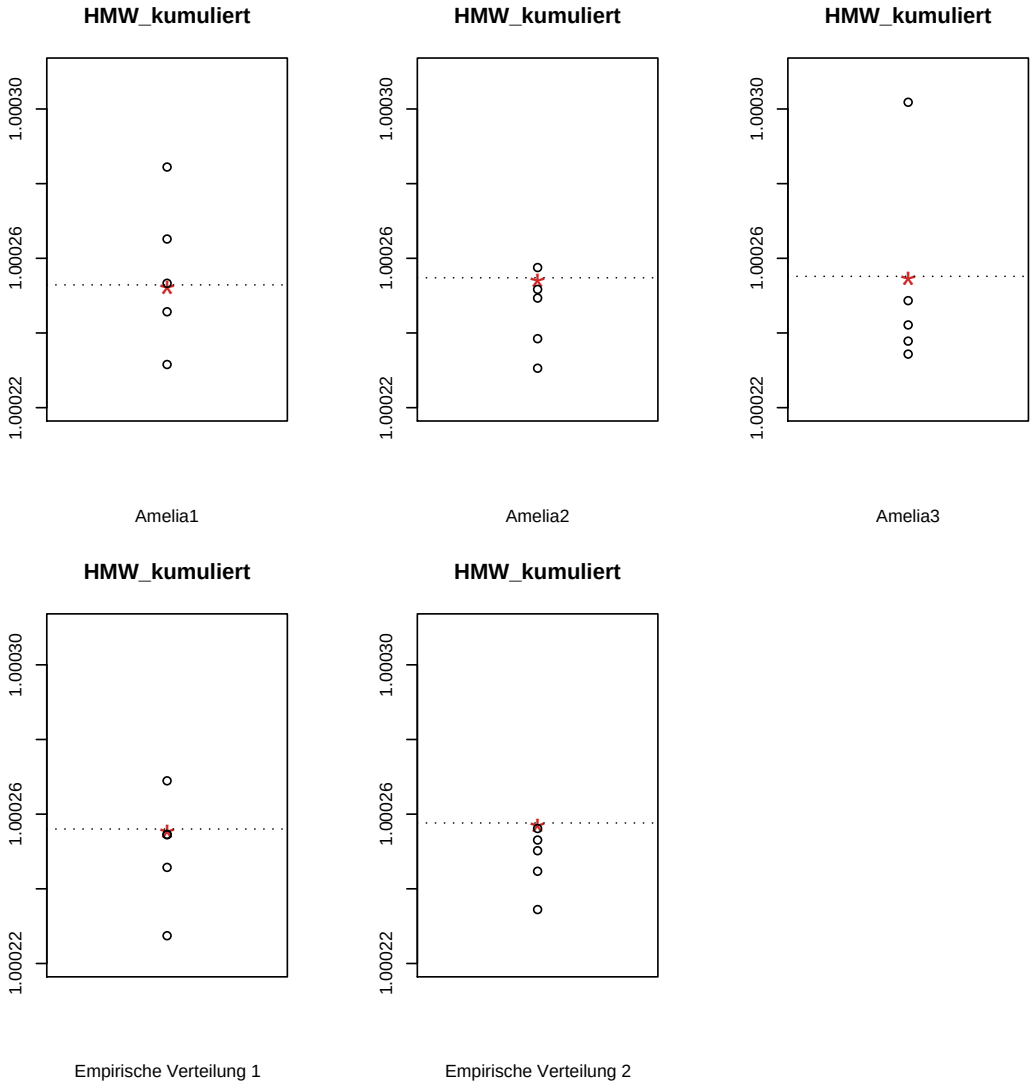


Abbildung B.9: Vergleich der Parameterschätzer - HMW_kumuliert

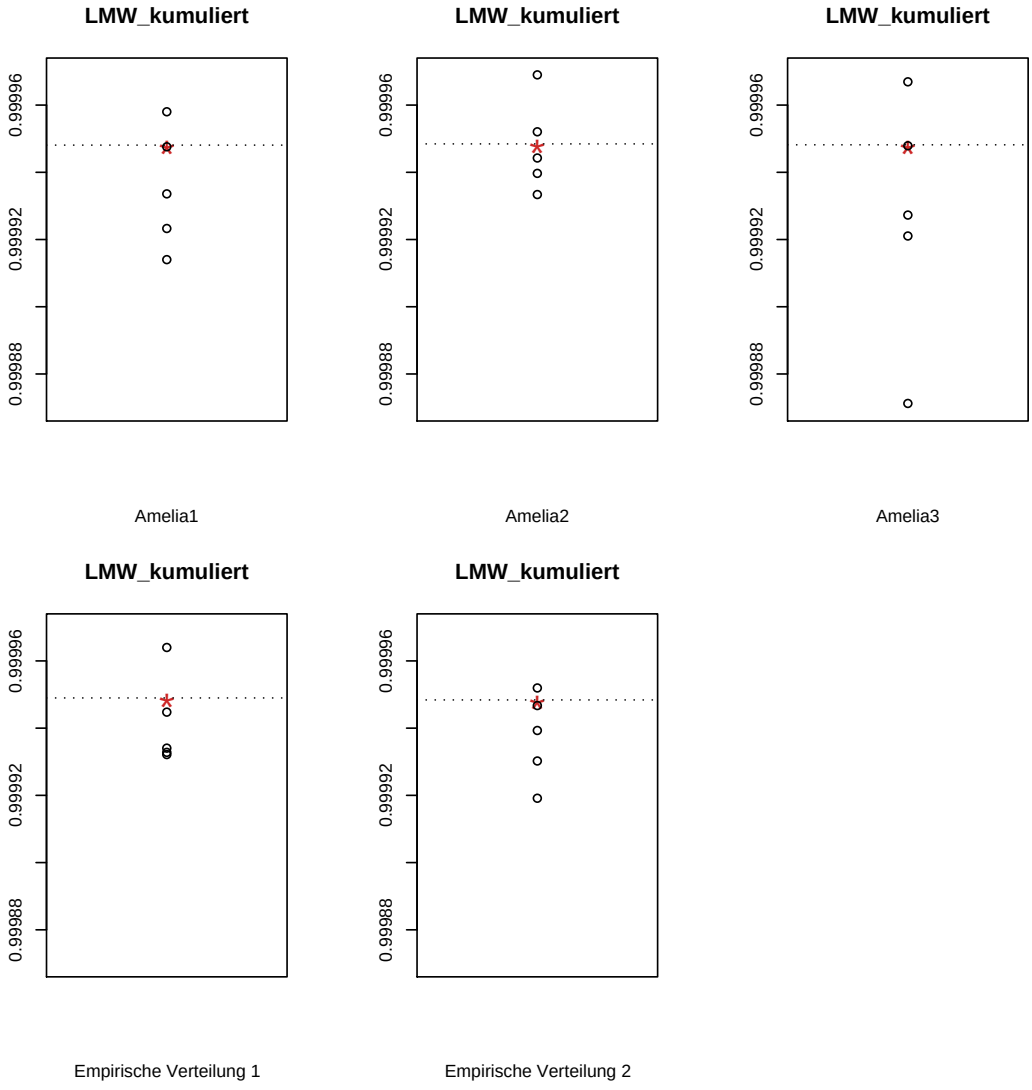


Abbildung B.10: Vergleich der Parameterschätzer - LMW_kumuliert

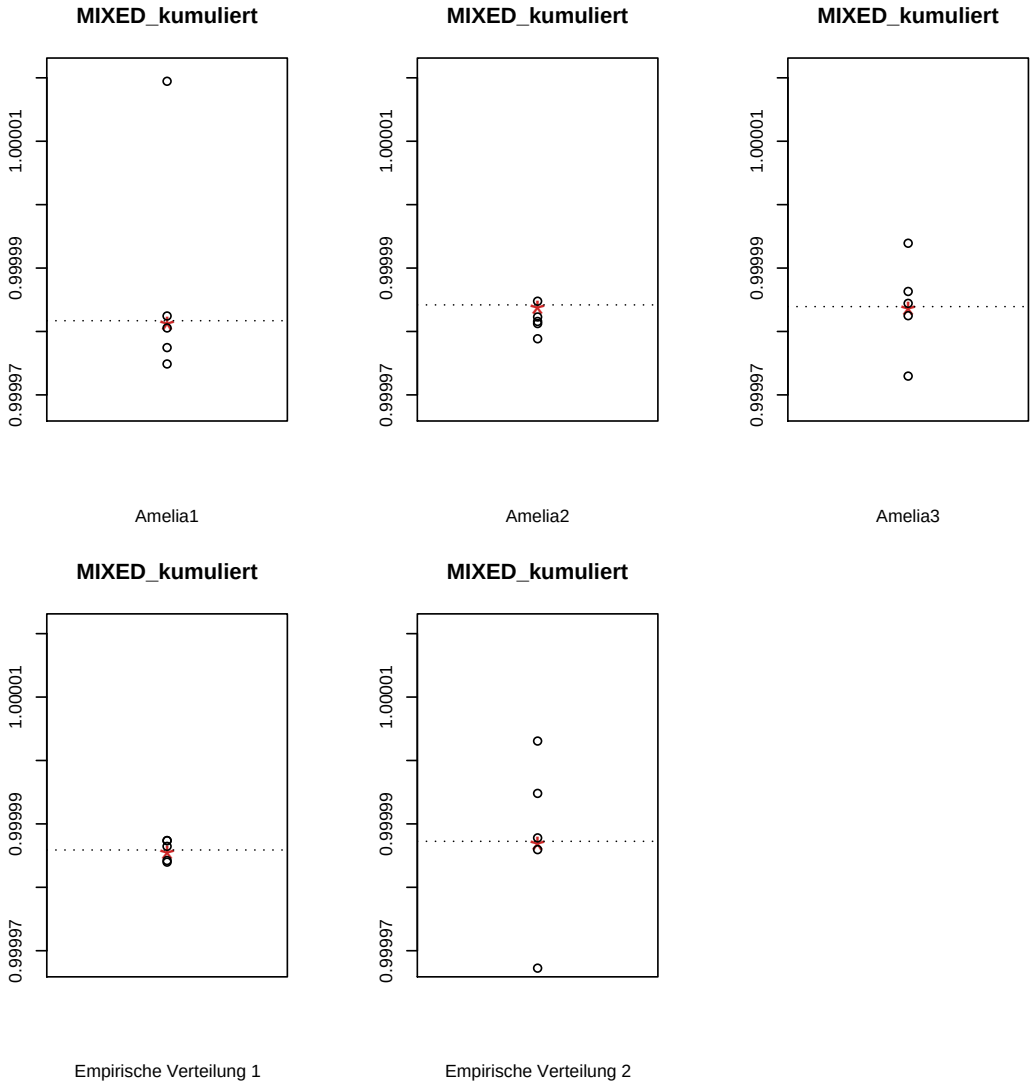


Abbildung B.11: Vergleich der Parameterschätzer - MIXED_kumuliert

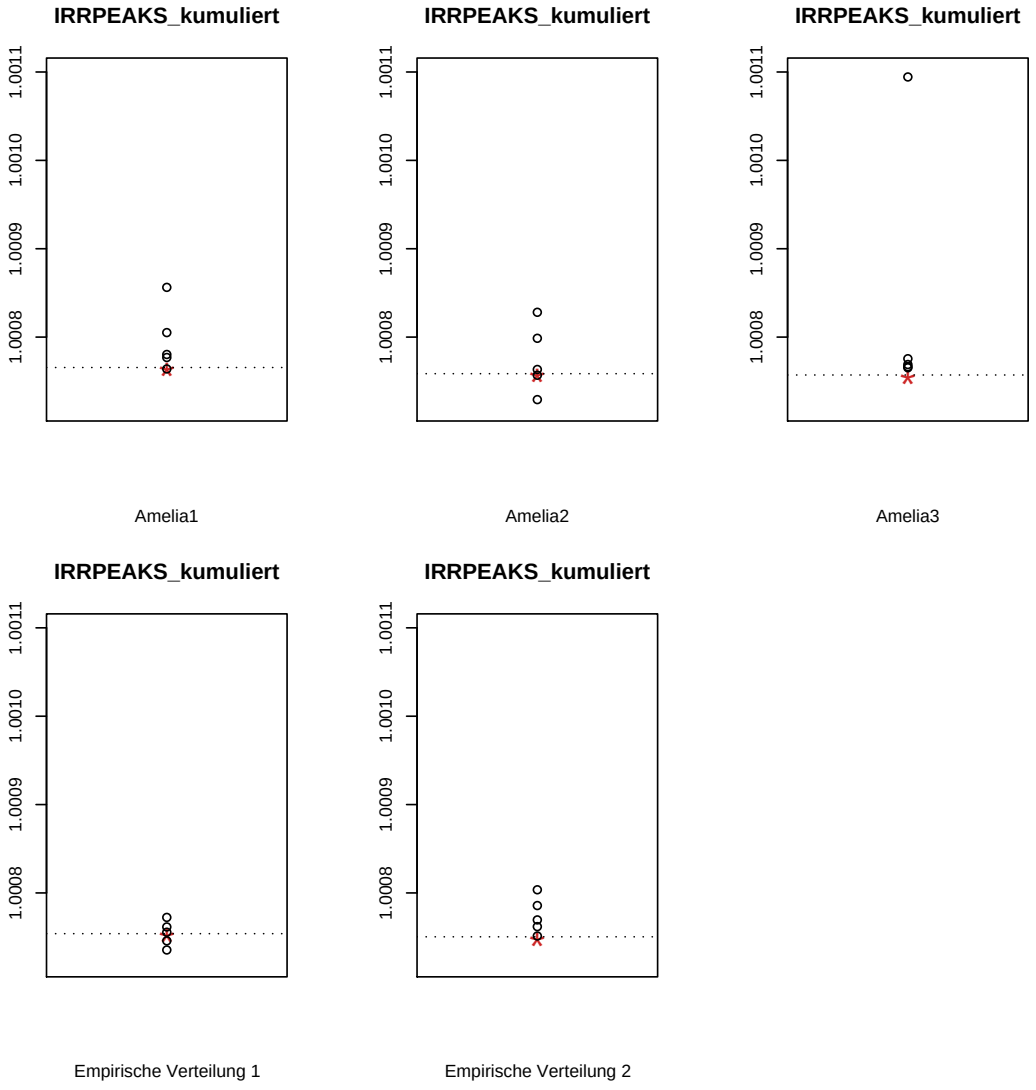


Abbildung B.12: Vergleich der Parameterschätzer - IRRPEAKS_kumuliert

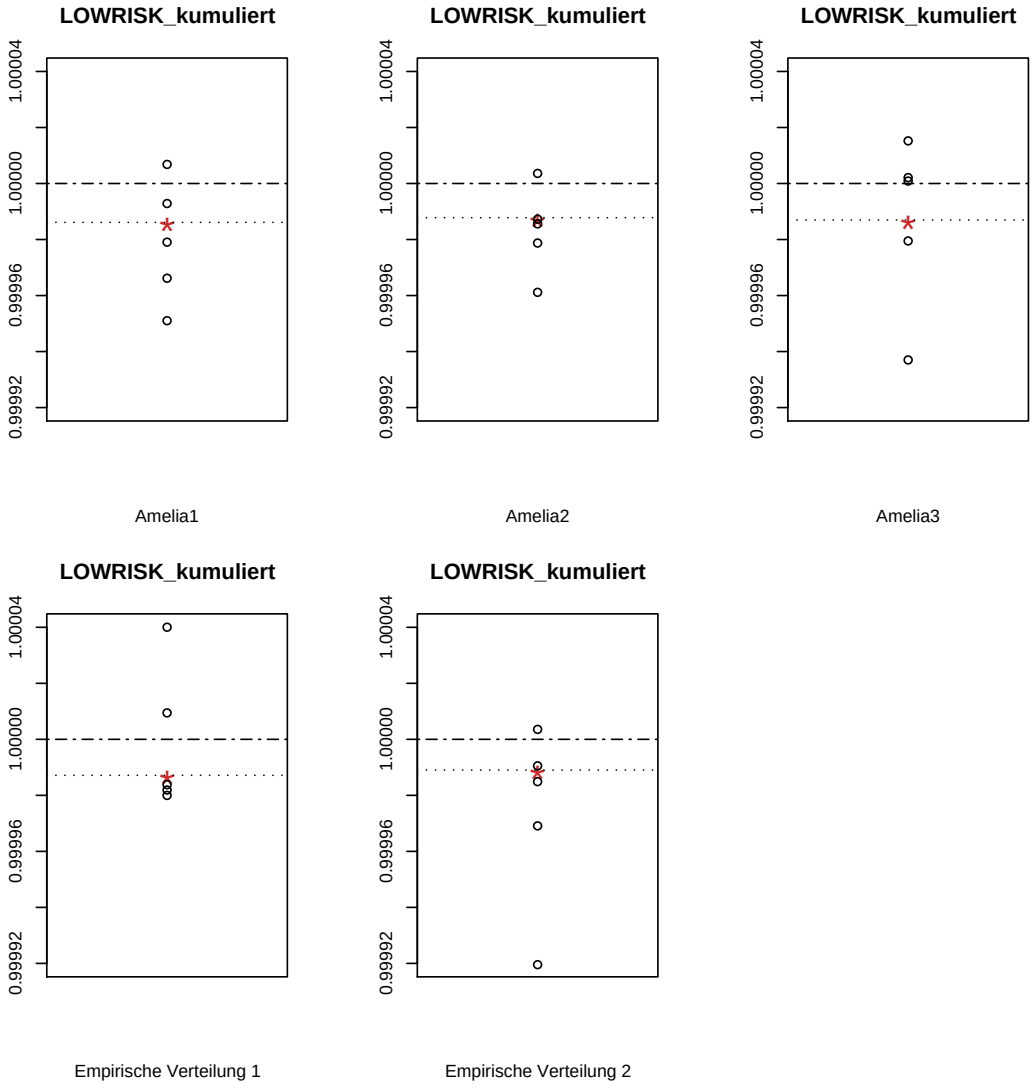


Abbildung B.13: Vergleich der Parameterschätzer - LOWRISK_kumuliert

ANHANG C

R-Code

C.1 Imputation der fehlenden Werte in den potentiellen Confoundervariablen

C.1.1 Imputation durch Ziehen gemäß der Randverteilung der Daten

```
> ###############################################################################
> # Imputation:Ziehen aus emp.Verteilung-Erstellung des 1.vervollst.Datensatzes  #
> ###############################################################################
>
> ###### 5 zufällige Startwerte auswählen #############
> set.seed(1)
> startwerte <- runif(2, min=1, max=1000)
> startwerte
> startwert1 <- round(startwerte[1])
> startwert1
> startwert2 <- round(startwerte[2])
> startwert2
> ###############################################################################
> ################## Imputation des  1. Datensatzes ############################
> ###############################################################################
>
> load("daten_fragebogen.RData")
> ###################
> # d_geb imputieren
> ###################
>
> n0 <- table(daten_fragebogen$d_geb, useNA="always")[[1]]
> n1 <- table(daten_fragebogen$d_geb, useNA="always")[[2]]
> n_miss <- table(daten_fragebogen$d_geb, useNA="always")[[3]]
> prob1 <- n1/(n0+n1)
> n0
> n1
> n_miss
> prob1
> set.seed(startwert1)  # muss vor jedem Befehl laufen, der Zufallszahlen zieht
> x <- rbinom(n=n_miss, size=1, prob=prob1)
> table(x)
> j <- 1
> for (i in 1:nrow(daten_fragebogen)){
+   if(is.na(daten_fragebogen$d_geb)[i]){
+     daten_fragebogen$d_geb[i] <- x[j]
+     j <- j+1
+   }
+ }
> table(daten_fragebogen$d_geb, useNA="always")
>
> ########################
> # PAR_ALL_r imputieren
```

```
> ########################
>
> n0 <- table(daten_fragebogen$PAR_ALL_r, useNA="always")[[1]]
> n1 <- table(daten_fragebogen$PAR_ALL_r, useNA="always")[[2]]
> n_miss <- table(daten_fragebogen$PAR_ALL_r, useNA="always")[[3]]
> prob1 <- n1/(n0+n1)
> n0
> n1
> n_miss
> prob1
> set.seed(startwert1)  # muss vor jedem Befehl laufen, der Zufallszahlen zieht
> x <- rbinom(n=n_miss, size=1, prob=prob1)
> table(x)
> j <- 1
> for (i in 1:nrow(daten_fragebogen)){
+   if(is.na(daten_fragebogen$PAR_ALL_r)[i]){
+     daten_fragebogen$PAR_ALL_r[i] <- x[j]
+     j <- j+1
+   }
+ }
> table(daten_fragebogen$PAR_ALL_r, useNA="always")
>
> ########################
> # siblings imputieren
> ########################
>
> n0 <- table(daten_fragebogen$siblings, useNA="always")[[1]]
> n1 <- table(daten_fragebogen$siblings, useNA="always")[[2]]
> n2 <- table(daten_fragebogen$siblings, useNA="always")[[3]]
> n3 <- table(daten_fragebogen$siblings, useNA="always")[[4]]
> n4 <- table(daten_fragebogen$siblings, useNA="always")[[5]]
> n5 <- table(daten_fragebogen$siblings, useNA="always")[[6]]
> n6 <- table(daten_fragebogen$siblings, useNA="always")[[7]]
> n7 <- table(daten_fragebogen$siblings, useNA="always")[[8]]
> n_miss <- table(daten_fragebogen$siblings, useNA="always")[[9]]
> prob0 <- n0/(n0+n1+n2+n3+n4+n5+n6+n7)
> prob1 <- n1/(n0+n1+n2+n3+n4+n5+n6+n7)
> prob2 <- n2/(n0+n1+n2+n3+n4+n5+n6+n7)
> prob3 <- n3/(n0+n1+n2+n3+n4+n5+n6+n7)
> prob4 <- n4/(n0+n1+n2+n3+n4+n5+n6+n7)
> prob5 <- n5/(n0+n1+n2+n3+n4+n5+n6+n7)
> prob6 <- n6/(n0+n1+n2+n3+n4+n5+n6+n7)
> prob7 <- n7/(n0+n1+n2+n3+n4+n5+n6+n7)
>
> set.seed(startwert1)  # muss vor jedem Befehl laufen, der Zufallszahlen zieht
> x <- sample(0:7, size=n_miss, replace=TRUE, prob=c(prob0,prob1,prob2,prob3,prob4,prob5,prob6,prob7))
> table(x)
> j <- 1
> for (i in 1:nrow(daten_fragebogen)){
+   if(is.na(daten_fragebogen$siblings)[i]){
+     daten_fragebogen$siblings[i] <- x[j]
+     j <- j+1
+   }
+ }
> table(daten_fragebogen$siblings, useNA="always")
>
> ####################
> # STILL imputieren
> ####################
>
> n0 <- table(daten_fragebogen$STILL_r, useNA="always")[[1]]
> n1 <- table(daten_fragebogen$STILL_r, useNA="always")[[2]]
> n_miss <- table(daten_fragebogen$STILL_r, useNA="always")[[3]]
> prob1 <- n1/(n0+n1)
> n0
> n1
> n_miss
> prob1
> set.seed(startwert1)  # muss vor jedem Befehl laufen, der Zufallszahlen zieht
> x <- rbinom(n=n_miss, size=1, prob=prob1)
> table(x)
> j <- 1
> for (i in 1:nrow(daten_fragebogen)){
```

```
+   if(is.na(daten_fragebogen$STILL_r)[i]){
+     daten_fragebogen$STILL_r[i] <- x[j]
+     j <- j+1
+   }
+ }
> table(daten_fragebogen$STILL_r, useNA="always")
>
> ####################
> # ETSNOW imputieren
> ####################
>
> n0 <- table(daten_fragebogen$ETSNOW_r, useNA="always")[[1]]
> n1 <- table(daten_fragebogen$ETSNOW_r, useNA="always")[[2]]
> n2 <- table(daten_fragebogen$ETSNOW_r, useNA="always")[[3]]
> n_miss <- table(daten_fragebogen$ETSNOW, useNA="always")[[4]] # Achtung: hier muss [[4]] stehen!
> prob0 <- n0/(n0+n1+n2)
> prob1 <- n1/(n0+n1+n2)
> prob2 <- n2/(n0+n1+n2)
> n0
> n1
> n2
> n_miss
> prob0
> prob1
> prob2
> set.seed(startwert1)  # muss vor jedem Befehl laufen, der Zufallszahlen zieht
> x <- sample(0:2, size=n_miss, replace=TRUE, prob=c(prob0,prob1,prob2))
> table(x)
> j <- 1
> for (i in 1:nrow(daten_fragebogen)){
+   if(is.na(daten_fragebogen$ETSNOW_r)[i]){
+     daten_fragebogen$ETSNOW_r[i] <- x[j]
+     j <- j+1
+   }
+ }
> table(daten_fragebogen$ETSNOW_r, useNA="always")
>
> ####################
> # f58x imputieren
> ####################
>
> n0 <- table(daten_fragebogen$f58x, useNA="always")[[1]]
> n1 <- table(daten_fragebogen$f58x, useNA="always")[[2]]
> n_miss <- table(daten_fragebogen$f58x, useNA="always")[[3]]
> prob1 <- n1/(n0+n1)
> n0
> n1
> n_miss
> prob1
> set.seed(startwert1)  # muss vor jedem Befehl laufen, der Zufallszahlen zieht
> x <- rbinom(n=n_miss, size=1, prob=prob1)
> table(x)
> j <- 1
> for (i in 1:nrow(daten_fragebogen)){
+   if(is.na(daten_fragebogen$f58x)[i]){
+     daten_fragebogen$f58x[i] <- x[j]
+     j <- j+1
+   }
+ }
> table(daten_fragebogen$f58x, useNA="always")
>
> ######################
> # RAUCHEN imputieren
> ######################
>
> n0 <- table(daten_fragebogen$RAUCHEN, useNA="always")[[1]]
> n1 <- table(daten_fragebogen$RAUCHEN, useNA="always")[[2]]
> n_miss <- table(daten_fragebogen$RAUCHEN, useNA="always")[[3]]
> prob1 <- n1/(n0+n1)
> n0
> n1
> n_miss
> prob1
```

```
> set.seed(startwert1)  # muss vor jedem Befehl laufen, der Zufallszahlen zieht
> x <- rbinom(n=n_miss, size=1, prob=prob1)
> table(x)
> j <- 1
> for (i in 1:nrow(daten_fragebogen)){
+   if(is.na(daten_fragebogen$RAUCHEN)[i]){
+     daten_fragebogen$RAUCHEN[i] <- x[j]
+     j <- j+1
+   }
+ }
> table(daten_fragebogen$RAUCHEN, useNA="always")
>
> ######################
> # s2f78 imputieren
> ######################
>
> n0 <- table(daten_fragebogen$s2f78, useNA="always")[[1]]
> n1 <- table(daten_fragebogen$s2f78, useNA="always")[[2]]
> n_miss <- table(daten_fragebogen$s2f78, useNA="always")[[3]]
> prob1 <- n1/(n0+n1)
> n0
> n1
> n_miss
> prob1
> set.seed(startwert1)  # muss vor jedem Befehl laufen, der Zufallszahlen zieht
> x <- rbinom(n=n_miss, size=1, prob=prob1)
> table(x)
> j <- 1
> for (i in 1:nrow(daten_fragebogen)){
+   if(is.na(daten_fragebogen$s2f78)[i]){
+     daten_fragebogen$s2f78[i] <- x[j]
+     j <- j+1
+   }
+ }
> table(daten_fragebogen$s2f78, useNA="always")
>
> ########################
> # s2RAUCHEN imputieren
> ########################
>
> n0 <- table(daten_fragebogen$s2RAUCHEN, useNA="always")[[1]]
> n1 <- table(daten_fragebogen$s2RAUCHEN, useNA="always")[[2]]
> n2 <- table(daten_fragebogen$s2RAUCHEN, useNA="always")[[3]]
> n_miss <- table(daten_fragebogen$s2RAUCHEN, useNA="always")[[4]]  # Achtung: hier muss [[4]] stehen!
> prob0 <- n0/(n0+n1+n2)
> prob1 <- n1/(n0+n1+n2)
> prob2 <- n2/(n0+n1+n2)
> n0
> n1
> n2
> n_miss
> prob0
> prob1
> prob2
> set.seed(startwert1)  # muss vor jedem Befehl laufen, der Zufallszahlen zieht
> x <- sample(0:2, size=n_miss, replace=TRUE, prob=c(prob0,prob1,prob2))
> table(x)
> j <- 1
> for (i in 1:nrow(daten_fragebogen)){
+   if(is.na(daten_fragebogen$s2RAUCHEN)[i]){
+     daten_fragebogen$s2RAUCHEN[i] <- x[j]
+     j <- j+1
+   }
+ }
> table(daten_fragebogen$s2RAUCHEN, useNA="always")
>
> ######################
> # s2SCHULE imputieren
> ######################
>
> n0 <- table(daten_fragebogen$s2SCHULE, useNA="always")[[1]]
> n1 <- table(daten_fragebogen$s2SCHULE, useNA="always")[[2]]
> n_miss <- table(daten_fragebogen$s2SCHULE, useNA="always")[[3]]
```

```
> prob1 <- n1/(n0+n1)
> n0
> n1
> n_miss
> prob1
> set.seed(startwert1)  # muss vor jedem Befehl laufen, der Zufallszahlen zieht
> x <- rbinom(n=n_miss, size=1, prob=prob1)
> table(x)
> j <- 1
> for (i in 1:nrow(daten_fragebogen)){
+   if(is.na(daten_fragebogen$s2SCHULE)[i]){
+     daten_fragebogen$s2SCHULE[i] <- x[j]
+     j <- j+1
+   }
+ }
> table(daten_fragebogen$s2SCHULE, useNA="always")
>
> ### Datensatz abspeichern
> empVert_imputed1 <- daten_fragebogen
> save(empVert_imputed1, file="empVert_imputed1.RData")
> load("empVert_imputed1.RData")
>
> ###################
> # SES_r imputieren (da es später auch für die Imputation der Berufsdaten bzw. für die Modelle benötigt wird)
> ###################
>
> n0 <- table(empVert_imputed1$SES_r, useNA="always")[[1]]
> n1 <- table(empVert_imputed1$SES_r, useNA="always")[[2]]
> n_miss <- table(empVert_imputed1$SES_r, useNA="always")[[3]]
> prob1 <- n1/(n0+n1)
> n0
> n1
> n_miss
> prob1
> set.seed(startwert1)  # muss vor jedem Befehl laufen, der Zufallszahlen zieht
> x <- rbinom(n=n_miss, size=1, prob=prob1)
> table(x)
> j <- 1
> for (i in 1:nrow(empVert_imputed1)){
+   if(is.na(empVert_imputed1$SES_r)[i]){
+     empVert_imputed1$SES_r[i] <- x[j]
+     j <- j+1
+   }
+ }
> table(empVert_imputed1$SES_r, useNA="always")
>
> ########################
> # BERUF imputieren
> ########################
>
> n1 <- table(empVert_imputed1$BERUF, useNA="always")[[1]]
> n2 <- table(empVert_imputed1$BERUF, useNA="always")[[2]]
> n3 <- table(empVert_imputed1$BERUF, useNA="always")[[3]]
> n4 <- table(empVert_imputed1$BERUF, useNA="always")[[4]]
> n5 <- table(empVert_imputed1$BERUF, useNA="always")[[5]]
> n6 <- table(empVert_imputed1$BERUF, useNA="always")[[6]]
> n7 <- table(empVert_imputed1$BERUF, useNA="always")[[7]]
> n9 <- table(empVert_imputed1$BERUF, useNA="always")[[8]]
> n12 <- table(empVert_imputed1$BERUF, useNA="always")[[9]]
> n_miss <- table(empVert_imputed1$BERUF, useNA="always")[[10]]
> prob1 <- n1/(n1+n2+n3+n4+n5+n6+n7+n9+n12)
> prob2 <- n2/(n1+n2+n3+n4+n5+n6+n7+n9+n12)
> prob3 <- n3/(n1+n2+n3+n4+n5+n6+n7+n9+n12)
> prob4 <- n4/(n1+n2+n3+n4+n5+n6+n7+n9+n12)
> prob5 <- n5/(n1+n2+n3+n4+n5+n6+n7+n9+n12)
> prob6 <- n6/(n1+n2+n3+n4+n5+n6+n7+n9+n12)
> prob7 <- n7/(n1+n2+n3+n4+n5+n6+n7+n9+n12)
> prob9 <- n9/(n1+n2+n3+n4+n5+n6+n7+n9+n12)
> prob12 <- n12/(n1+n2+n3+n4+n5+n6+n7+n9+n12)
>
> set.seed(startwert1)  # muss vor jedem Befehl laufen, der Zufallszahlen zieht
> x <- sample(c(1,2,3,4,5,6,7,9,12), size=n_miss, replace=TRUE, prob=c(prob1,prob2,prob3,prob4,prob5,prob6,prob7,prob9,prob12))
> table(x)
```

```
> j <- 1
> for (i in 1:nrow(empVert_imputed1)){
+   if(is.na(empVert_imputed1$BERUF)[i]){
+     empVert_imputed1$BERUF[i] <- x[j]
+     j <- j+1
+   }
+ }
> table(empVert_imputed1$BERUF, useNA="always")
>
> ########################
> # s2BERUF imputieren
> ########################
>
> n1 <- table(empVert_imputed1$s2BERUF, useNA="always")[[1]]
> n2 <- table(empVert_imputed1$s2BERUF, useNA="always")[[2]]
> n3 <- table(empVert_imputed1$s2BERUF, useNA="always")[[3]]
> n4 <- table(empVert_imputed1$s2BERUF, useNA="always")[[4]]
> n5 <- table(empVert_imputed1$s2BERUF, useNA="always")[[5]]
> n6 <- table(empVert_imputed1$s2BERUF, useNA="always")[[6]]
> n8 <- table(empVert_imputed1$s2BERUF, useNA="always")[[7]]
> n9 <- table(empVert_imputed1$s2BERUF, useNA="always")[[8]]
> n_miss <- table(empVert_imputed1$s2BERUF, useNA="always")[[9]]
> prob1 <- n1/(n1+n2+n3+n4+n5+n6+n8+n9)
> prob2 <- n2/(n1+n2+n3+n4+n5+n6+n8+n9)
> prob3 <- n3/(n1+n2+n3+n4+n5+n6+n8+n9)
> prob4 <- n4/(n1+n2+n3+n4+n5+n6+n8+n9)
> prob5 <- n5/(n1+n2+n3+n4+n5+n6+n8+n9)
> prob6 <- n6/(n1+n2+n3+n4+n5+n6+n8+n9)
> prob8 <- n8/(n1+n2+n3+n4+n5+n6+n8+n9)
> prob9 <- n9/(n1+n2+n3+n4+n5+n6+n8+n9)
> set.seed(startwert1)  # muss vor jedem Befehl laufen, der Zufallszahlen zieht
> x <- sample(c(1,2,3,4,5,6,8,9), size=n_miss, replace=TRUE, prob=c(prob1,prob2,prob3,prob4,prob5,prob6,prob8,prob9))
> table(x)
> j <- 1
> for (i in 1:nrow(empVert_imputed1)){
+   if(is.na(empVert_imputed1$s2BERUF)[i]){
+     empVert_imputed1$s2BERUF[i] <- x[j]
+     j <- j+1
+   }
+ }
> table(empVert_imputed1$s2BERUF, useNA="always")
>
> ### Datensatz abspeichern
> save(empVert_imputed1, file="empVert_imputed1.RData")
>
> Bei der Imputation des 2.Datensatzes wurde analog vorgegangen
```

C.1.2 Imputation mithilfe des R-Packages AMELIA II

```
> ##############################################################################
> ### Imputation mit Amelia - Erstellung von 3 vervollständigten Datensätzen  ####
> ##############################################################################
> library(Amelia)
> load("daten_fragebogen.RData")
> ##############################################################################
> ########### Korrelationen prüfen wg. Problem der Multikollinearität ############
> ##############################################################################
>
> # Bravais-Pearson-Koeffizient für metrische und binäre Daten
> cor(daten_fragebogen,use="pairwise.complete.obs")
> # Rangkorrelationskoeffizient für kategoriale Daten
> cor(daten_fragebogen,use="pairwise.complete.obs",method="spearman")
> #pearson:
> #CUR_HAY_r(1,2) und CURHAYV(0,1): -0.42
> #CURHAYV(0,1) und s2CURHAYV(0,1): 0.56
> #CUR_ASTH_r(1,2) und CURASTHV(0,1): -0.44
> #CURASTHV(0,1) und s2CURASTHV(0,1): 0.48
> #CUR_DERM_r(1,2) und CURDERMV(0,1): -0.58
>
> #spearman:
> #RAUCHEN(0,1) und s2RAUCHEN(0,1,2): 0.62
>
```

```
> ##############################################################################
> ###################### Imputation mit AMELIA-II  ##############################
> ##############################################################################
>
> # noms = ... Nominale Variablen angeben; (ords=... Ordinale Variablen angeben)
> # idvars=knr (wird nicht bei Imputation verwendet, bleibt aber im Datensatz)
> # Kollinearität ebenfalls bei RAUCHEN und s2RAUCHEN, trotzdem ins Modell aufnehmen
>
> set.seed(123)# für Reproduzierbarkeit
> amelia_all <- amelia(daten_fragebogen, m = 5, idvars=c("knr"), noms=c("zentrum","f02x",
+ "d_geb","PAR_ALL_r","siblings","STILL_r","ETSNOW_r","SES_r","f58x","RAUCHEN","BERUF",
+ "s2f78","s2SCHULE","s2RAUCHEN","s2BERUF","CUR_DERM_r","CUR_HAY_r","CUR_ASTH_r",
+ "CURDERMV","CURHAYV","CURASTHV","s2CURHAYV","s2CURASTHV"),empri=5,9)
>
> ################################
> #einzelne Datensätze abspeichern
> ################################
> amelia_imputed1 <- amelia_all$m1
> save(amelia_imputed1, file="amelia_imputed1.RData")
> amelia_imputed2 <- amelia_all$m2
> save(amelia_imputed2, file="amelia_imputed2.RData")
> amelia_imputed3 <- amelia_all$m3
> save(amelia_imputed3, file="amelia_imputed3.RData")
```

C.2 Berechnung der Expositionsvariablen

```
> ##############################################################################
> ################### Datensatz laden ##########################################
> ##############################################################################
>
> # Für die Job-Matrix werden nur die Probanden mit vollständigen Berufsdaten
> # verwendet
> load("berufsdaten_vollstaendig.RData") #-> aus Tinn-R Datei berufsdaten_alle.r
> nrow(berufsdaten_vollstaendig) # enthält 10940 Zeilen also 1094 Probanden
> # (weil pro Proband 10 Zeilen für maximal 10 Jobs)
>
> basis <- berufsdaten_vollstaendig
> ##############################################################################
> ################### Basisdatensatz erstellen #################################
> ##############################################################################
>
> # JEM auf 5 Kategorien (HMW, LMW, MIXED, IRRPEAKS, LOWRISK) reduzieren
> # Dabei steht jeweils in der Obergruppe (z.B. HMW) eine 1 (für exponiert), wenn
> # in mindestens einer Unterkategorie (z.B. anim, fish, flour ... für HMW) eine 1
> # (für exponiert) steht
> basis$HMW <- 0
> basis$LMW <- 0
> basis$MIXED <- 0
> basis$IRRPEAKS <- 0
> basis$LOWRISK <- 0
> # anim, fish, flour, plants, mites, enzymes, latex, bioaero, drugs zur
> # Obergruppe HMW zusammenfassen
> basis$HMW[(basis$anim==1) | (basis$fish==1) | (basis$flour==1) |
+ (basis$plants==1) | (basis$mites==1) | (basis$enzymes==1) | (basis$latex==1) |
+ (basis$bioaero==1) | (basis$drugs==1)] <-1
> # react, isocy, clean, wood, metals zur Obergruppe LMW zusammenfassen
> basis$LMW[(basis$react==1) | (basis$isocy==1) | (basis$clean==1) |
+ (basis$wood==1) | (basis$metals==1)] <-1
> # mwf, textile, agric zur Obergruppe MIXED zusammenfassen
> basis$MIXED[(basis$mwf==1) | (basis$textile==1) | (basis$agric==1)] <-1
> # IRRPEAKS (Obergruppe) kann direkt von irrpeaks(Untergruppe) übernommen werden,
> # es gibt hier keine anderen Untergruppen
> basis$IRRPEAKS[(basis$irrpeaks==1)] <-1
> # exhaust, ets, pos_irr zur Obergruppe LOWRISK zusammenfassen
> basis$LOWRISK[(basis$exhaust==1) | (basis$ets==1) | (basis$pos_irr==1) |
+ (basis$low_anti==1)] <-1
> # basis umbennen in basis_5kat und # die Daten abspeichern - die Daten enthalten
> # noch ALLE vollständigen Probanden
> basis_5kat <- basis
> #save(basis_5kat, file="basis_5kat.RData")
>
```

```
> # Sortierung entsprechend vornehmen, so dass 1. Beruf in der ersten Zeile steht etc.
> # Sortieren nach knr, dann nach ANF_JAHR und wenn Jahr gleich ist nach ANF_MONAT
> # => der erste Beruf steht immer in der ersten Zeile
> # wurde bereits in daten_beruf_alle_sort vorgenommen!!!
> basis_5kat_sort <- basis_5kat
> # Variable NR_BERUF anlegen, gibt an in welcher Reihenfolge die Berufe ausgeübt
> # wurden
> basis_5kat_sort$NR_BERUF <- rep(1:10, times=nrow(basis)/10)
> # Nur die Spalten die auch wirklich benötigt werden als Subset rausziehen
> expoberechnung_basis <- subset(basis_5kat_sort, select=c("knr", "NR_BERUF","ANF_MONAT","ANF_JAHR",
+ "END_MONATx","END_JAHRx","WST","ISCO","HMW", "LMW","MIXED","IRRPEAKS","LOWRISK","DAUER","JEMALS_GEARB"))
> # Für spätere Berechnungen: DAUER, WST wenn == NA auf 0 setzten
> # WST == NA: das sind die Fälle mit ISCO 94, 95, 97, 98 (da dürfen die WST fehlen)
> # DAUER == NA: das sind die Fälle mit WST < 8 (da dürfen die WST fehlen!)
> # oder die Zeilen ohne Berufsangaben (also mit ISCO 8888 bzw. 9999)
> expoberechnung_basis$DAUER[is.na(expoberechnung_basis$DAUER)] <- 0
> expoberechnung_basis$WST[is.na(expoberechnung_basis$WST)] <- 0
> # Variable MIND_8_WST nochmal neu anlegen damit überall 0 oder 1 drin steht
> expoberechnung_basis$MIND_8_WST <- 0
> expoberechnung_basis$MIND_8_WST[expoberechnung_basis$WST>=8] <- 1
> # Jetzt die Exposition pro Beruf (d.h. für jede Zeile) berechnen
>
> # Erst mal alle Variablen mit 0 initialisieren
> expoberechnung_basis$HMW_beruf <- 0
> expoberechnung_basis$LMW_beruf <- 0
> expoberechnung_basis$MIXED_beruf <- 0
> expoberechnung_basis$IRRPEAKS_beruf <- 0
> expoberechnung_basis$LOWRISK_beruf <- 0
> # Expositionen nur berechnen wenn MIND_8_WST == 1 ist
> # für ISCO 94, 95, 97, 98 ist sowieso überall Expo == 0 (passt also)
> for (i in 1:nrow(expoberechnung_basis)){
+ if (expoberechnung_basis$MIND_8_WST[i] == 1){
+ expoberechnung_basis$HMW_beruf[i] <- (4.25 * expoberechnung_basis$WST[i] *
+ expoberechnung_basis$HMW[i] * expoberechnung_basis$DAUER[i])
+ expoberechnung_basis$LMW_beruf[i] <- (4.25 * expoberechnung_basis$WST[i] *
+ expoberechnung_basis$LMW[i] * expoberechnung_basis$DAUER[i])
+ expoberechnung_basis$MIXED_beruf[i] <- (4.25 * expoberechnung_basis$WST[i] *
+ expoberechnung_basis$MIXED[i] * expoberechnung_basis$DAUER[i])
+ expoberechnung_basis$IRRPEAKS_beruf[i] <- (4.25 * expoberechnung_basis$WST[i] *
+ expoberechnung_basis$IRRPEAKS[i] * expoberechnung_basis$DAUER[i])
+ expoberechnung_basis$LOWRISK_beruf[i] <- (4.25 * expoberechnung_basis$WST[i] *
+ expoberechnung_basis$LOWRISK[i] * expoberechnung_basis$DAUER[i])
+ }
+ }
> # Datensatz abspeichern
> save(expoberechnung_basis, file="expoberechnung_basis.RData")
> ##############################################################################
> ###################### Exposition kumuliert ##################################
> ##############################################################################
>
> # Jetzt über die Jobs aufsummieren => pro Proband eine Zeile
> expo_kumuliert <- data.frame(HMW_kumuliert=numeric(nrow(basis)/10),
+ LMW_kumuliert=numeric(nrow(basis)/10), MIXED_kumuliert=numeric(nrow(basis)/10),
+ IRRPEAKS_kumuliert=numeric(nrow(basis)/10), LOWRISK_kumuliert=numeric(nrow(basis)/10),
+ HMW_binaer=numeric(nrow(basis)/10), LMW_binaer=numeric(nrow(basis)/10), MIXED_binaer=numeric(nrow(basis)/10),
+ IRRPEAKS_binaer=numeric(nrow(basis)/10), LOWRISK_binaer=numeric(nrow(basis)/10))
> # alle initialisieren mit NA
> expo_kumuliert$HMW_kumuliert <- NA
> expo_kumuliert$LMW_kumuliert <- NA
> expo_kumuliert$MIXED_kumuliert <- NA
> expo_kumuliert$IRRPEAKS_kumuliert <- NA
> expo_kumuliert$LOWRISK_kumuliert <- NA
> expo_kumuliert$HMW_binaer <- NA
> expo_kumuliert$LMW_binaer <- NA
> expo_kumuliert$MIXED_binaer <- NA
> expo_kumuliert$IRRPEAKS_binaer <- NA
> expo_kumuliert$LOWRISK_binaer <- NA
> # KNR übertragen
> i <- 1
> j <- 1
> while (i <= nrow(expoberechnung_basis)){
+ expo_kumuliert$knr[j] <- as.character(expoberechnung_basis$knr[i])
+ i <- i+10 # nächster Proband
```

```
+ j <- j+1  # nächster Proband in der neuen Matrix
+ }
> # HMW_beruf pro Proband über die 10 Jobs aufsummieren
> i <- 1
> j <- 1
> while (i <= nrow(expoberechnung_basis)){
+ expo_kumuliert$HMW_kumuliert[j] <- (
+ expoberechnung_basis$HMW_beruf[i] +
+ expoberechnung_basis$HMW_beruf[i+1] +
+ expoberechnung_basis$HMW_beruf[i+2] +
+ expoberechnung_basis$HMW_beruf[i+3] +
+ expoberechnung_basis$HMW_beruf[i+4] +
+ expoberechnung_basis$HMW_beruf[i+5] +
+ expoberechnung_basis$HMW_beruf[i+6] +
+ expoberechnung_basis$HMW_beruf[i+7] +
+ expoberechnung_basis$HMW_beruf[i+8] +
+ expoberechnung_basis$HMW_beruf[i+9])
+ i <- i+10  # nächster Proband in expoberechnung_basis
+ j <- j+1   # nächste Stelle in der neuen Matrix expo_kumuliert
+ }
> # binäre Variable erstellen - gibt an ob Exposition bestand oder nicht
> expo_kumuliert$HMW_binaer[expo_kumuliert$HMW_kumuliert > 0] <- 1
> expo_kumuliert$HMW_binaer[expo_kumuliert$HMW_kumuliert == 0] <- 0
> # LMW_beruf pro Proband über die 10 Jobs aufsummieren
> i <- 1
> j <- 1
> while (i <= nrow(expoberechnung_basis)){
+ expo_kumuliert$LMW_kumuliert[j] <- (
+ expoberechnung_basis$LMW_beruf[i] +
+ expoberechnung_basis$LMW_beruf[i+1] +
+ expoberechnung_basis$LMW_beruf[i+2] +
+ expoberechnung_basis$LMW_beruf[i+3] +
+ expoberechnung_basis$LMW_beruf[i+4] +
+ expoberechnung_basis$LMW_beruf[i+5] +
+ expoberechnung_basis$LMW_beruf[i+6] +
+ expoberechnung_basis$LMW_beruf[i+7] +
+ expoberechnung_basis$LMW_beruf[i+8] +
+ expoberechnung_basis$LMW_beruf[i+9])
+ i <- i+10  # nächster Proband in expoberechnung_basis
+ j <- j+1   # nächste Stelle in der neuen Matrix expo_kumuliert
+ }
> # binäre Variable erstellen - gibt an ob Exposition bestand oder nicht
> expo_kumuliert$LMW_binaer[expo_kumuliert$LMW_kumuliert > 0] <- 1
> expo_kumuliert$LMW_binaer[expo_kumuliert$LMW_kumuliert == 0] <- 0
> # MIXED_beruf pro Proband über die 10 Jobs aufsummieren
> i <- 1
> j <- 1
> while (i <= nrow(expoberechnung_basis)){
+ expo_kumuliert$MIXED_kumuliert[j] <- (
+ expoberechnung_basis$MIXED_beruf[i] +
+ expoberechnung_basis$MIXED_beruf[i+1] +
+ expoberechnung_basis$MIXED_beruf[i+2] +
+ expoberechnung_basis$MIXED_beruf[i+3] +
+ expoberechnung_basis$MIXED_beruf[i+4] +
+ expoberechnung_basis$MIXED_beruf[i+5] +
+ expoberechnung_basis$MIXED_beruf[i+6] +
+ expoberechnung_basis$MIXED_beruf[i+7] +
+ expoberechnung_basis$MIXED_beruf[i+8] +
+ expoberechnung_basis$MIXED_beruf[i+9])
+ i <- i+10  # nächster Proband in expoberechnung_basis
+ j <- j+1   # nächste Stelle in der neuen Matrix expo_kumuliert
+ }
> # binäre Variable erstellen - gibt an ob Exposition bestand oder nicht
> expo_kumuliert$MIXED_binaer[expo_kumuliert$MIXED_kumuliert > 0] <- 1
> expo_kumuliert$MIXED_binaer[expo_kumuliert$MIXED_kumuliert == 0] <- 0
> # IRRPEAKS_beruf pro Proband über die 10 Jobs aufsummieren
> i <- 1
> j <- 1
> while (i <= nrow(expoberechnung_basis)){
+ expo_kumuliert$IRRPEAKS_kumuliert[j] <- (
+ expoberechnung_basis$IRRPEAKS_beruf[i] +
+ expoberechnung_basis$IRRPEAKS_beruf[i+1] +
+ expoberechnung_basis$IRRPEAKS_beruf[i+2] +
```

```
+ expoberechnung_basis$IRRPEAKS_beruf[i+3] +
+ expoberechnung_basis$IRRPEAKS_beruf[i+4] +
+ expoberechnung_basis$IRRPEAKS_beruf[i+5] +
+ expoberechnung_basis$IRRPEAKS_beruf[i+6] +
+ expoberechnung_basis$IRRPEAKS_beruf[i+7] +
+ expoberechnung_basis$IRRPEAKS_beruf[i+8] +
+ expoberechnung_basis$IRRPEAKS_beruf[i+9])
+ i <- i+10  # nächster Proband in expoberechnung_basis
+ j <- j+1   # nächste Stelle in der neuen Matrix expo_kumuliert
+ }
> # binäre Variable erstellen - gibt an ob Exposition bestand oder nicht
> expo_kumuliert$IRRPEAKS_binaer[expo_kumuliert$IRRPEAKS_kumuliert > 0] <- 1
> expo_kumuliert$IRRPEAKS_binaer[expo_kumuliert$IRRPEAKS_kumuliert == 0] <- 0
> # LOWRISK_beruf pro Proband über die 10 Jobs aufsummieren
> i <- 1
> j <- 1
> while (i <= nrow(expoberechnung_basis)){
+ expo_kumuliert$LOWRISK_kumuliert[j] <- (
+ expoberechnung_basis$LOWRISK_beruf[i] +
+ expoberechnung_basis$LOWRISK_beruf[i+1] +
+ expoberechnung_basis$LOWRISK_beruf[i+2] +
+ expoberechnung_basis$LOWRISK_beruf[i+3] +
+ expoberechnung_basis$LOWRISK_beruf[i+4] +
+ expoberechnung_basis$LOWRISK_beruf[i+5] +
+ expoberechnung_basis$LOWRISK_beruf[i+6] +
+ expoberechnung_basis$LOWRISK_beruf[i+7] +
+ expoberechnung_basis$LOWRISK_beruf[i+8] +
+ expoberechnung_basis$LOWRISK_beruf[i+9])
+ i <- i+10  # nächster Proband in expoberechnung_basis
+ j <- j+1   # nächste Stelle in der neuen Matrix expo_kumuliert
+ }
> # binäre Variable erstellen - gibt an ob Exposition bestand oder nicht
> expo_kumuliert$LOWRISK_binaer[expo_kumuliert$LOWRISK_kumuliert > 0] <- 1
> expo_kumuliert$LOWRISK_binaer[expo_kumuliert$LOWRISK_kumuliert == 0] <- 0
> # Datensatz abspeichern
> save(expo_kumuliert, file="expo_kumuliert.RData")
> ##############################################################################
> ###             Die Exposition im ersten Beruf berechnen                   ###
> ##############################################################################
>
> load("expoberechnung_basis.RData")
> expo_ersterberuf_basis <- subset(expoberechnung_basis, NR_BERUF == 1)
> # Datensatz abspeichern
> #save(expo_ersterberuf_basis, file="expo_ersterberuf_basis.RData")
>
> expo_ersterberuf_basis$HMW_ersterberuf_gesamt <- expo_ersterberuf_basis$HMW_beruf
> expo_ersterberuf_basis$LMW_ersterberuf_gesamt <- expo_ersterberuf_basis$LMW_beruf
> expo_ersterberuf_basis$MIXED_ersterberuf_gesamt <- expo_ersterberuf_basis$MIXED_beruf
> expo_ersterberuf_basis$IRRPEAKS_ersterberuf_gesamt <- expo_ersterberuf_basis$IRRPEAKS_beruf
> expo_ersterberuf_basis$LOWRISK_ersterberuf_gesamt <- expo_ersterberuf_basis$LOWRISK_beruf
> # binäre Variable erstellen - gibt an ob Exposition bestand oder nicht
> expo_ersterberuf_basis$HMW_ersterberuf_binaer[expo_ersterberuf_basis$HMW_ersterberuf_gesamt > 0] <- 1
> expo_ersterberuf_basis$HMW_ersterberuf_binaer[expo_ersterberuf_basis$HMW_ersterberuf_gesamt == 0] <- 0
> # binäre Variable erstellen - gibt an ob Exposition bestand oder nicht
> expo_ersterberuf_basis$LMW_ersterberuf_binaer[expo_ersterberuf_basis$LMW_ersterberuf_gesamt > 0] <- 1
> expo_ersterberuf_basis$LMW_ersterberuf_binaer[expo_ersterberuf_basis$LMW_ersterberuf_gesamt == 0] <- 0
> # binäre Variable erstellen - gibt an ob Exposition bestand oder nicht
> expo_ersterberuf_basis$MIXED_ersterberuf_binaer[expo_ersterberuf_basis$MIXED_ersterberuf_gesamt > 0] <- 1
> expo_ersterberuf_basis$MIXED_ersterberuf_binaer[expo_ersterberuf_basis$MIXED_ersterberuf_gesamt == 0] <- 0
> # binäre Variable erstellen - gibt an ob Exposition bestand oder nicht
> expo_ersterberuf_basis$IRRPEAKS_ersterberuf_binaer[expo_ersterberuf_basis$IRRPEAKS_ersterberuf_gesamt > 0] <- 1
> expo_ersterberuf_basis$IRRPEAKS_ersterberuf_binaer[expo_ersterberuf_basis$IRRPEAKS_ersterberuf_gesamt == 0] <- 0
> # binäre Variable erstellen - gibt an ob Exposition bestand oder nicht
> expo_ersterberuf_basis$LOWRISK_ersterberuf_binaer[expo_ersterberuf_basis$LOWRISK_ersterberuf_gesamt > 0] <- 1
> expo_ersterberuf_basis$LOWRISK_ersterberuf_binaer[expo_ersterberuf_basis$LOWRISK_ersterberuf_gesamt == 0] <- 0
> # nur die relevanten Variablen bleiben im Datensatz
> expo_ersterberuf <- subset(expo_ersterberuf_basis, select=c("knr", "HMW_ersterberuf_gesamt",
+ "LMW_ersterberuf_gesamt","MIXED_ersterberuf_gesamt", "IRRPEAKS_ersterberuf_gesamt","LOWRISK_ersterberuf_gesamt",
+ "HMW_ersterberuf_binaer","LMW_ersterberuf_binaer","MIXED_ersterberuf_binaer",
+ "IRRPEAKS_ersterberuf_binaer","LOWRISK_ersterberuf_binaer"))
> # Datensatz abspeichern
> save(expo_ersterberuf, file="expo_ersterberuf.RData")
>
```

```
> ##############################################################################
> ###             Die Exposition im ersten Berufsjahr berechnen                ###
> ##############################################################################
>
> load("expoberechnung_basis.RData")
> expo_erstesjahr_basis <- expoberechnung_basis
> # Hier dürfen nur die ISCOs 94,95,97,98 vorkommen oder WST sind kleiner als 8
> a<-subset(expo_erstesjahr_basis, is.na(ANF_MONAT) & ISCO!=8888 & ISCO!=9999)
> #fix(a)
> b<-subset(expo_erstesjahr_basis, is.na(ANF_JAHR) & ISCO!=8888 & ISCO!=9999)
> #fix(b)
>
> # Variable ANF_BERUF_CHAR erstellen: in ihr sollen ANF_JAHR und ANF_MONAT in
> # folgender Form zusammengefasst werden:"JJJJMM" also zb. "200103" für März 2001
>
> # Variable ANF_BERUF_CHAR mit 0 initialisieren
> expo_erstesjahr_basis$ANF_BERUF_CHAR <- 0
> # ANF_BERUF_CHAR wird nur da eingetragen, wo ANF_MONAT und ANF_JAHR vorhanden
> # sind
> for (i in 1:nrow(expo_erstesjahr_basis)){
+ if (!is.na(expo_erstesjahr_basis$ANF_MONAT[i]) & (!is.na(expo_erstesjahr_basis$ANF_JAHR[i]))){
+ # Wenn der ANF_MONAT kleiner als 10 ist, dann muss das Format so aussehen:
+ # "JJJJOM" weil z.B. der Monat März im numerischen "3" ist und nicht "03"
+ if (expo_erstesjahr_basis$ANF_MONAT[i] < 10){
+ expo_erstesjahr_basis$ANF_BERUF_CHAR[i]<-paste(expo_erstesjahr_basis$ANF_JAHR[i]
+ ,"0",expo_erstesjahr_basis$ANF_MONAT[i],sep="")
+ }
+ # Wenn der ANF_MONAT größer oder gleich 10 ist,dann muss das Format so aussehen:
+ # "JJJJMM", d.h. man kann Jahr und Monat einfach hintereinander zusammenfügen
+ if (expo_erstesjahr_basis$ANF_MONAT[i] >= 10){
+ expo_erstesjahr_basis$ANF_BERUF_CHAR[i]<-paste(expo_erstesjahr_basis$ANF_JAHR[i]
+ ,expo_erstesjahr_basis$ANF_MONAT[i],sep="")
+ }
+ }
+ }
> # Variable ENDE_12_MONATE erstellen: in ihr steht das Ende der ersten 12 Monate
> # im Beruf (also des ersten Berufsjahres), d.h. zum Beginn des ersten Berufs
> # eines Probanden werden 12 Monate addiert, dann hat man den Endzeitpunkt des
> # ersten Berufsjahres (wieder im Format "JJJJMM" also zb. "200103" für März 2001
> # (so wie bei ANF_BERUF_CHAR))
>
> # Variable ENDE_12_MONATE mit 0 initialisieren
> expo_erstesjahr_basis$ENDE_12_MONATE <- 0
> # Wenn ANF_BERUF_CHAR ungleich 0 ist und NR_BERUF = 1, d.h. beim ersten Beruf
> # jedes Probanden, wenn dort ein Beginn steht (1. Beruf ist immer in der ersten
> # Zeile da zuvor sortiert wurde), soll ein ENDE_12_MONATE berechnet werden
> for (i in 1:nrow(expo_erstesjahr_basis)){
+ if (expo_erstesjahr_basis$ANF_BERUF_CHAR[i]!=0 & expo_erstesjahr_basis$ANF_BERUF_CHAR[i]!="000000"
+ & expo_erstesjahr_basis$NR_BERUF[i]==1){
+ # Wenn der ANF_MONAT 1 ist, dann ist das ENDE_12_MONATE im gleichen Jahr wie
+ # ANF_BERUF_CHAR, d.h. das Jahr kann so übernommen werden; beim Monat müssen 11
+ # Monate zum ANF_MONAT dazu addiert werden. Bsp: ANF_BERUF_CHAR ist 200001, dann
+ # ist ENDE_12_MONATE 200012 (genau ein Jahr = 12 Monate)
+ if (expo_erstesjahr_basis$ANF_MONAT[i]==1){ende12monate <- paste(expo_erstesjahr_basis$ANF_JAHR[i],
+ expo_erstesjahr_basis$ANF_MONAT[i]+11,sep="")
+ expo_erstesjahr_basis$ENDE_12_MONATE[i] <- ende12monate
+ expo_erstesjahr_basis$ENDE_12_MONATE[i+1] <- ende12monate
+ expo_erstesjahr_basis$ENDE_12_MONATE[i+2] <- ende12monate
+ expo_erstesjahr_basis$ENDE_12_MONATE[i+3] <- ende12monate
+ expo_erstesjahr_basis$ENDE_12_MONATE[i+4] <- ende12monate
+ expo_erstesjahr_basis$ENDE_12_MONATE[i+5] <- ende12monate
+ expo_erstesjahr_basis$ENDE_12_MONATE[i+6] <- ende12monate
+ expo_erstesjahr_basis$ENDE_12_MONATE[i+7] <- ende12monate
+ expo_erstesjahr_basis$ENDE_12_MONATE[i+8] <- ende12monate
+ expo_erstesjahr_basis$ENDE_12_MONATE[i+9] <- ende12monate
+ }
+ # Wenn der Anfang Monat ungleich 1 ist
+ if (expo_erstesjahr_basis$ANF_MONAT[i]!=1){
+ # Wenn der ANF_MONAT 11 oder 12 ist, dann wird zum Jahr in ANF_BERUF_CHAR noch
+ # ein Jahr addiert (Ende der 12 Monate liegt im nächsten Jahr); Vom ANF_MONAT
+ # muss ein Monat abgezogen werden. Bsp: ANF_BERUF_CHAR ist 200211, dann ist
+ # ENDE_12_MONATE 200310.
+ if (expo_erstesjahr_basis$ANF_MONAT[i] > 10){ende12monate <- paste(expo_erstesjahr_basis$ANF_JAHR[i]+1,
```

```
+ expo_erstesjahr_basis$ANF_MONAT[i]-1,sep="")
+ expo_erstesjahr_basis$ENDE_12_MONATE[i] <- ende12monate
+ expo_erstesjahr_basis$ENDE_12_MONATE[i+1] <- ende12monate
+ expo_erstesjahr_basis$ENDE_12_MONATE[i+2] <- ende12monate
+ expo_erstesjahr_basis$ENDE_12_MONATE[i+3] <- ende12monate
+ expo_erstesjahr_basis$ENDE_12_MONATE[i+4] <- ende12monate
+ expo_erstesjahr_basis$ENDE_12_MONATE[i+5] <- ende12monate
+ expo_erstesjahr_basis$ENDE_12_MONATE[i+6] <- ende12monate
+ expo_erstesjahr_basis$ENDE_12_MONATE[i+7] <- ende12monate
+ expo_erstesjahr_basis$ENDE_12_MONATE[i+8] <- ende12monate
+ expo_erstesjahr_basis$ENDE_12_MONATE[i+9] <- ende12monate
+
+ }
+ # Wenn der ANF_MONAT kleiner gleich 10 ist (2,3,...,10), dann wird zum Jahr in
+ # ANF_BERUF_CHAR noch ein Jahr addiert (Ende der 12 Monate liegt im nächsten
+ # Jahr); Vom ANF_MONAT muss ein Monat abgezogen werden, da dann ANF_MONAT-1 in
+ # Menge (1,2,...,9) liegt muss zwischen JAHR und ANF_MONAT-1 noch eine "0"
+ # eingefügt werden. Bsp: ANF_BERUF_CHAR ist 200210, dann ist ENDE_12_MONATE
+ # 200309.
+ if (expo_erstesjahr_basis$ANF_MONAT[i] <= 10){ende12monate <- paste(expo_erstesjahr_basis$ANF_JAHR[i]+1,"0",
+ expo_erstesjahr_basis$ANF_MONAT[i]-1,sep="")
+ expo_erstesjahr_basis$ENDE_12_MONATE[i] <- ende12monate
+ expo_erstesjahr_basis$ENDE_12_MONATE[i+1] <- ende12monate
+ expo_erstesjahr_basis$ENDE_12_MONATE[i+2] <- ende12monate
+ expo_erstesjahr_basis$ENDE_12_MONATE[i+3] <- ende12monate
+ expo_erstesjahr_basis$ENDE_12_MONATE[i+4] <- ende12monate
+ expo_erstesjahr_basis$ENDE_12_MONATE[i+5] <- ende12monate
+ expo_erstesjahr_basis$ENDE_12_MONATE[i+6] <- ende12monate
+ expo_erstesjahr_basis$ENDE_12_MONATE[i+7] <- ende12monate
+ expo_erstesjahr_basis$ENDE_12_MONATE[i+8] <- ende12monate
+ expo_erstesjahr_basis$ENDE_12_MONATE[i+9] <- ende12monate
+ }
+ }
+ }
+ }
> # Indikatorvariable BERUF_BEACHTEN erstellen, die angibt, ob der Beruf in der
> # enstprechenden Zeile für die Exposition im ersten Berufsjahr berücksichtigt
> # werden muss oder nicht (0=nein /1=ja)
> expo_erstesjahr_basis$BERUF_BEACHTEN <- 0
> for(i in 1:nrow(expo_erstesjahr_basis)){if(expo_erstesjahr_basis$ANF_BERUF_CHAR[i]!=0 &
+ (expo_erstesjahr_basis$ANF_BERUF_CHAR[i] < expo_erstesjahr_basis$ENDE_12_MONATE[i])){
+ expo_erstesjahr_basis$BERUF_BEACHTEN[i] <- 1
+ }
+ }
> # Variable ENDE_12_MONATE auftrennen in ENDE_12_MONATE_MONAT und
> # ENDE_12_MONATE_JAHR. Bsp: Wenn ENDE_12_MONATE = "200103" ist dann steht jetzt
> # in ENDE_12_MONATE_MONAT "3" und in ENDE_12_MONATE_JAHR "2001".
>
> # Beide Variablen ENDE_12_MONATE_MONAT und ENDE_12_MONATE_JAHR mit NA initialis.
> expo_erstesjahr_basis$ENDE_12_MONATE_MONAT <- NA
> expo_erstesjahr_basis$ENDE_12_MONATE_JAHR <- NA
> # ENDE_12_MONATE_MONAT steht in ENDE_12_MONATE an den Stellen 5-6 (substring)
> # ENDE_12_MONATE_JAHR steht in ENDE_12_MONATE an den Stellen 1-4 (substring)
> for(i in 1:nrow(expo_erstesjahr_basis)){
+ if(expo_erstesjahr_basis$ENDE_12_MONATE[i]!=0){
+ expo_erstesjahr_basis$ENDE_12_MONATE_MONAT[i] <- as.numeric(
+ substring(expo_erstesjahr_basis$ENDE_12_MONATE[i], 5, 6))
+ expo_erstesjahr_basis$ENDE_12_MONATE_JAHR[i] <- as.numeric(
+ substring(expo_erstesjahr_basis$ENDE_12_MONATE[i], 1, 4))
+ }
+ }
> # Jetzt soll bei den Berufen die für das erste Berufsjahr berücksichtigt werden
> # müssen (BERUF_BEACHTEN = 1) berechnet werden, wieviele Monate noch in den
> # 12-Monats-Zeitraum fallen. Bsp: wenn der Beruf "200103" beginnt und das
> # ENDE_12_MONATE ist "200105", dann fallen 3 Monate noch in den 12-Monats-
> # Zeitraum (es wird jeweils einschließlich Anfangs- und Endmonat gerechnet) !
>
> # Variable DIFF_MONAT mit NA initialisieren
> expo_erstesjahr_basis$DIFF_MONAT <- NA
> for(i in 1:nrow(expo_erstesjahr_basis)){
+ if(expo_erstesjahr_basis$BERUF_BEACHTEN[i]==1){
+ # Wenn das ANF_JAHR gleich dem ENDE_12_MONATE_JAHR ist, dann ist die DIFF_MONAT:
+ # (ENDE_12_MONATE_MONAT - ANF_MONAT +1); ist also die Anzahl der Monate die für
```

```
+ # den jeweiligen Beruf für das erste Berufsjahr noch berücksichtigt werden
+ # müssen
+ if(expo_erstesjahr_basis$ANF_JAHR[i] == expo_erstesjahr_basis$ENDE_12_MONATE_JAHR[i]){
+ expo_erstesjahr_basis$DIFF_MONAT[i] <- (expo_erstesjahr_basis$ENDE_12_MONATE_MONAT[i] -
+ expo_erstesjahr_basis$ANF_MONAT[i] + 1)
+ }
+ # Wenn das ANF_JAHR kleiner ist als das ENDE_12_MONATE_JAHR, dann ist die
+ # DIFF_MONAT: ((12 - ANF_MONAT) + ENDE_12_MONATE_MONAT +1); ist also die Anzahl
+ # der Monate die für den jeweiligen Beruf für das erste Berufsjahr noch
+ # berücksichtigt werden müssen
+ if(expo_erstesjahr_basis$ANF_JAHR[i] < expo_erstesjahr_basis$ENDE_12_MONATE_JAHR[i]){
+ expo_erstesjahr_basis$DIFF_MONAT[i] <-((12 - expo_erstesjahr_basis$ANF_MONAT[i])
+ + expo_erstesjahr_basis$ENDE_12_MONATE_MONAT[i] + 1)
+ }
+ }
+ }
> # Jetzt wird noch abgeglichen, wieviele Monate in dem Beruf insgesamt gearbeitet
> # wurden (DAUER) und wieviele Monate noch in den 12-Monats-Zeitraum fallen
> # (DIFF_MONAT)
>
> # Variable MONATE_BEACHTEN mit 0 initialisieren
> expo_erstesjahr_basis$MONATE_BEACHTEN <- 0
> # Nur für die Fälle abgleichen, bei denen DIFF_MONAT nicht NA ist
> for(i in 1:nrow(expo_erstesjahr_basis)){
+ if(!is.na(expo_erstesjahr_basis$DIFF_MONAT[i])){
+ # Wenn die Anzahl der Monate die im Beruf gearbeitet wurden (DAUER)
+ # größer ist als die Anzahl der Monate, die noch in den 12-Monats-Zeitraum
+ # fallen (DIFF_MONAT), dann ist die Anzahl der Monate, die für das erste
+ # Berufsjahr noch beachtet werden muss (MONATE_BEACHTEN) gleich DIFF_MONAT
+ # Wenn die Anzahl der Monate die im Beruf gearbeitet wurden (DAUER)
+ # kleiner gleich der Anzahl der Monate, die noch in den 12-Monats-Zeitraum
+ # fallen (DIFF_MONAT) ist, dann ist die Anzahl der Monate, die für das
+ # erste Berufsjahr noch beachtet werden muss (MONATE_BEACHTEN) gleich
+ # DAUER;
+ # => MONATE_BEACHTEN entspricht also dem Minimum von DAUER und DIFF_MONAT
+ expo_erstesjahr_basis$MONATE_BEACHTEN[i] <- min(
+ expo_erstesjahr_basis$DIFF_MONAT[i],expo_erstesjahr_basis$DAUER[i])}
+ }
> # Wo BERUF_BEACHTEN==1 und MIND_8_WST == 1 wird Expo pro Zeile berechnet, sonst 0
>
> # zunächst Variablen mit 0 initialisieren
> expo_erstesjahr_basis$HMW_beruf_erstesjahr <- 0
> expo_erstesjahr_basis$LMW_beruf_erstesjahr <- 0
> expo_erstesjahr_basis$MIXED_beruf_erstesjahr <- 0
> expo_erstesjahr_basis$IRRPEAKS_beruf_erstesjahr <- 0
> expo_erstesjahr_basis$LOWRISK_beruf_erstesjahr <- 0
> # Wo BERUF_BEACHTEN==1 und MIND_8_WST == 1 wird Expo pro Zeile berechnet
> for(i in 1:nrow(expo_erstesjahr_basis)){if((expo_erstesjahr_basis$BERUF_BEACHTEN[i]==1)
+ &(expo_erstesjahr_basis$MIND_8_WST[i]==1)){
+ expo_erstesjahr_basis$HMW_beruf_erstesjahr[i]  <- (4.25 *
+ expo_erstesjahr_basis$WST[i] * expo_erstesjahr_basis$HMW[i] *
+ expo_erstesjahr_basis$MONATE_BEACHTEN[i])
+ expo_erstesjahr_basis$LMW_beruf_erstesjahr[i]  <- (4.25 *
+ expo_erstesjahr_basis$WST[i] * expo_erstesjahr_basis$LMW[i] *
+ expo_erstesjahr_basis$MONATE_BEACHTEN[i])
+ expo_erstesjahr_basis$MIXED_beruf_erstesjahr[i]  <- (4.25 *
+ expo_erstesjahr_basis$WST[i] * expo_erstesjahr_basis$MIXED[i] *
+ expo_erstesjahr_basis$MONATE_BEACHTEN[i])
+ expo_erstesjahr_basis$IRRPEAKS_beruf_erstesjahr[i]  <- (4.25 *
+ expo_erstesjahr_basis$WST[i] * expo_erstesjahr_basis$IRRPEAKS[i] *
+ expo_erstesjahr_basis$MONATE_BEACHTEN[i])
+ expo_erstesjahr_basis$LOWRISK_beruf_erstesjahr[i]  <- (4.25 *
+ expo_erstesjahr_basis$WST[i] * expo_erstesjahr_basis$LOWRISK[i] *
+ expo_erstesjahr_basis$MONATE_BEACHTEN[i])
+ }
+ }
> # Jetzt über die Jobs aufsummieren
> # pro Proband eine Zeile
> expo_erstesjahr <- data.frame(HMW_erstesjahr_gesamt=numeric(nrow(basis)/10),
+ LMW_erstesjahr_gesamt=numeric(nrow(basis)/10), MIXED_erstesjahr_gesamt=numeric(nrow(basis)/10),
+ IRRPEAKS_erstesjahr_gesamt=numeric(nrow(basis)/10),
+ LOWRISK_erstesjahr_gesamt=numeric(nrow(basis)/10), HMW_erstesjahr_binaer=numeric(nrow(basis)/10),
+ LMW_erstesjahr_binaer=numeric(nrow(basis)/10), MIXED_erstesjahr_binaer=numeric(nrow(basis)/10),
```

```
+ IRRPEAKS_erstesjahr_binaer=numeric(nrow(basis)/10),
+ LOWRISK_erstesjahr_binaer=numeric(nrow(basis)/10))
> # alle Variablen initialisieren mit NA
> expo_erstesjahr$HMW_erstesjahr_gesamt <- NA
> expo_erstesjahr$LMW_erstesjahr_gesamt <- NA
> expo_erstesjahr$MIXED_erstesjahr_gesamt <- NA
> expo_erstesjahr$IRRPEAKS_erstesjahr_gesamt <- NA
> expo_erstesjahr$LOWRISK_erstesjahr_gesamt <- NA
> expo_erstesjahr$HMW_erstesjahr_binaer <- NA
> expo_erstesjahr$LMW_erstesjahr_binaer <- NA
> expo_erstesjahr$MIXED_erstesjahr_binaer <- NA
> expo_erstesjahr$IRRPEAKS_erstesjahr_binaer <- NA
> expo_erstesjahr$LOWRISK_erstesjahr_binaer <- NA
> # KNR übertragen
> i <- 1
> j <- 1
> while (i <= nrow(expo_erstesjahr_basis)){
+ expo_erstesjahr$knr[j] <- as.character(expo_erstesjahr_basis$knr[i])
+ i <- i+10 # nächster Proband
+ j <- j+1  # nächster Proband in der neuen Matrix
+ }
> # HMW_beruf_erstesjahr pro Proband über die 10 Jobs aufsummieren
> i <- 1
> j <- 1
> while (i <= nrow(expo_erstesjahr_basis)){
+ expo_erstesjahr$HMW_erstesjahr_gesamt[j] <- (
+ expo_erstesjahr_basis$HMW_beruf_erstesjahr[i] +
+ expo_erstesjahr_basis$HMW_beruf_erstesjahr[i+1] +
+ expo_erstesjahr_basis$HMW_beruf_erstesjahr[i+2] +
+ expo_erstesjahr_basis$HMW_beruf_erstesjahr[i+3] +
+ expo_erstesjahr_basis$HMW_beruf_erstesjahr[i+4] +
+ expo_erstesjahr_basis$HMW_beruf_erstesjahr[i+5] +
+ expo_erstesjahr_basis$HMW_beruf_erstesjahr[i+6] +
+ expo_erstesjahr_basis$HMW_beruf_erstesjahr[i+7] +
+ expo_erstesjahr_basis$HMW_beruf_erstesjahr[i+8] +
+ expo_erstesjahr_basis$HMW_beruf_erstesjahr[i+9])
+ i <- i+10  # nächster Proband in expo_erstesjahr_basis
+ j <- j+1   # nächste Stelle in der neuen Matrix expo_erstesjahr
+ }
> # binäre Variable erstellen - gibt an ob Exposition bestand oder nicht
> expo_erstesjahr$HMW_erstesjahr_binaer[expo_erstesjahr$HMW_erstesjahr_gesamt > 0] <- 1
> expo_erstesjahr$HMW_erstesjahr_binaer[expo_erstesjahr$HMW_erstesjahr_gesamt == 0] <- 0
> # LMW_beruf_erstesjahr pro Proband über die 10 Jobs aufsummieren
> i <- 1
> j <- 1
> while (i <= nrow(expo_erstesjahr_basis)){
+ expo_erstesjahr$LMW_erstesjahr_gesamt[j] <- (
+ expo_erstesjahr_basis$LMW_beruf_erstesjahr[i] +
+ expo_erstesjahr_basis$LMW_beruf_erstesjahr[i+1] +
+ expo_erstesjahr_basis$LMW_beruf_erstesjahr[i+2] +
+ expo_erstesjahr_basis$LMW_beruf_erstesjahr[i+3] +
+ expo_erstesjahr_basis$LMW_beruf_erstesjahr[i+4] +
+ expo_erstesjahr_basis$LMW_beruf_erstesjahr[i+5] +
+ expo_erstesjahr_basis$LMW_beruf_erstesjahr[i+6] +
+ expo_erstesjahr_basis$LMW_beruf_erstesjahr[i+7] +
+ expo_erstesjahr_basis$LMW_beruf_erstesjahr[i+8] +
+ expo_erstesjahr_basis$LMW_beruf_erstesjahr[i+9])
+ i <- i+10  # nächster Proband in expo_erstesjahr_basis
+ j <- j+1   # nächste Stelle in der neuen Matrix expo_erstesjahr
+ }
> # binäre Variable erstellen - gibt an ob Exposition bestand oder nicht
> expo_erstesjahr$LMW_erstesjahr_binaer[expo_erstesjahr$LMW_erstesjahr_gesamt > 0] <- 1
> expo_erstesjahr$LMW_erstesjahr_binaer[expo_erstesjahr$LMW_erstesjahr_gesamt == 0] <- 0
> # MIXED_beruf_erstesjahr pro Proband über die 10 Jobs aufsummieren
> i <- 1
> j <- 1
> while (i <= nrow(expo_erstesjahr_basis)){
+ expo_erstesjahr$MIXED_erstesjahr_gesamt[j] <- (
+ expo_erstesjahr_basis$MIXED_beruf_erstesjahr[i] +
+ expo_erstesjahr_basis$MIXED_beruf_erstesjahr[i+1] +
+ expo_erstesjahr_basis$MIXED_beruf_erstesjahr[i+2] +
+ expo_erstesjahr_basis$MIXED_beruf_erstesjahr[i+3] +
+ expo_erstesjahr_basis$MIXED_beruf_erstesjahr[i+4] +
```

```
+ expo_erstesjahr_basis$MIXED_beruf_erstesjahr[i+5] +
+ expo_erstesjahr_basis$MIXED_beruf_erstesjahr[i+6] +
+ expo_erstesjahr_basis$MIXED_beruf_erstesjahr[i+7] +
+ expo_erstesjahr_basis$MIXED_beruf_erstesjahr[i+8] +
+ expo_erstesjahr_basis$MIXED_beruf_erstesjahr[i+9])
+ i <- i+10  # nächster Proband in expo_erstesjahr_basis
+ j <- j+1   # nächste Stelle in der neuen Matrix expo_erstesjahr
+ }
> # binäre Variable erstellen - gibt an ob Exposition bestand oder nicht
> expo_erstesjahr$MIXED_erstesjahr_binaer[expo_erstesjahr$MIXED_erstesjahr_gesamt > 0] <- 1
> expo_erstesjahr$MIXED_erstesjahr_binaer[expo_erstesjahr$MIXED_erstesjahr_gesamt == 0] <- 0
> # IRRPEAKS_beruf_erstesjahr pro Proband über die 10 Jobs aufsummieren
> i <- 1
> j <- 1
> while (i <= nrow(expo_erstesjahr_basis)){
+ expo_erstesjahr$IRRPEAKS_erstesjahr_gesamt[j] <- (
+ expo_erstesjahr_basis$IRRPEAKS_beruf_erstesjahr[i] +
+ expo_erstesjahr_basis$IRRPEAKS_beruf_erstesjahr[i+1] +
+ expo_erstesjahr_basis$IRRPEAKS_beruf_erstesjahr[i+2] +
+ expo_erstesjahr_basis$IRRPEAKS_beruf_erstesjahr[i+3] +
+ expo_erstesjahr_basis$IRRPEAKS_beruf_erstesjahr[i+4] +
+ expo_erstesjahr_basis$IRRPEAKS_beruf_erstesjahr[i+5] +
+ expo_erstesjahr_basis$IRRPEAKS_beruf_erstesjahr[i+6] +
+ expo_erstesjahr_basis$IRRPEAKS_beruf_erstesjahr[i+7] +
+ expo_erstesjahr_basis$IRRPEAKS_beruf_erstesjahr[i+8] +
+ expo_erstesjahr_basis$IRRPEAKS_beruf_erstesjahr[i+9])
+ i <- i+10  # nächster Proband in expo_erstesjahr_basis
+ j <- j+1   # nächste Stelle in der neuen Matrix expo_erstesjahr
+ }
> # binäre Variable erstellen - gibt an ob Exposition bestand oder nicht
> expo_erstesjahr$IRRPEAKS_erstesjahr_binaer[expo_erstesjahr$IRRPEAKS_erstesjahr_gesamt > 0] <- 1
> expo_erstesjahr$IRRPEAKS_erstesjahr_binaer[expo_erstesjahr$IRRPEAKS_erstesjahr_gesamt == 0] <- 0
> # LOWRISK_beruf_erstesjahr pro Proband über die 10 Jobs aufsummieren
> i <- 1
> j <- 1
> while (i <= nrow(expo_erstesjahr_basis)){
+ expo_erstesjahr$LOWRISK_erstesjahr_gesamt[j] <- (
+ expo_erstesjahr_basis$LOWRISK_beruf_erstesjahr[i] +
+ expo_erstesjahr_basis$LOWRISK_beruf_erstesjahr[i+1] +
+ expo_erstesjahr_basis$LOWRISK_beruf_erstesjahr[i+2] +
+ expo_erstesjahr_basis$LOWRISK_beruf_erstesjahr[i+3] +
+ expo_erstesjahr_basis$LOWRISK_beruf_erstesjahr[i+4] +
+ expo_erstesjahr_basis$LOWRISK_beruf_erstesjahr[i+5] +
+ expo_erstesjahr_basis$LOWRISK_beruf_erstesjahr[i+6] +
+ expo_erstesjahr_basis$LOWRISK_beruf_erstesjahr[i+7] +
+ expo_erstesjahr_basis$LOWRISK_beruf_erstesjahr[i+8] +
+ expo_erstesjahr_basis$LOWRISK_beruf_erstesjahr[i+9])
+ i <- i+10  # nächster Proband in expo_erstesjahr_basis
+ j <- j+1   # nächste Stelle in der neuen Matrix expo_erstesjahr
+ }
> # binäre Variable erstellen - gibt an ob Exposition bestand oder nicht
> expo_erstesjahr$LOWRISK_erstesjahr_binaer[expo_erstesjahr$LOWRISK_erstesjahr_gesamt > 0] <- 1
> expo_erstesjahr$LOWRISK_erstesjahr_binaer[expo_erstesjahr$LOWRISK_erstesjahr_gesamt == 0] <- 0
> # Datensatz abspeichern
> save(expo_erstesjahr, file="expo_erstesjahr.RData")
```

C.3 Logistische Regression

Für das logistische Modell für die Zielgröße Allergische Rhinitis in SOLAR II, Schritte 1-5 beispielhaft für einen Datensatz in dem die Counfoundervariablen mit dem R-Package AMELIA II imputiert wurden. Im 6. Schritt werden die Schätzer aller 5 Datensätze kombiniert.

C.3.1 Schritt 1 - Confoundermodell

```
> ##############################################################################
> ########## Logit-Modell - Datensatz Amelia 1 ###################################
> ##############################################################################
> load("basis_modell_HEUSCHNUPFEN_amelia1.RData")
>
```

```
> ###############################################################################
> # Confounder-Modell aus Fragebogenvariablen auswählen #
> ###############################################################################
> HAY_confounder <- glm(s2CURHAYV ~ zentrum_r + d_geb + PAR_ALL_r + GESCHW +
+ STILL_r + CUR_DERM_r + CUR_HAY_r + ETSNOW_r + f02x + CURDERMV + CURHAYV + f58x
+ + RAUCHEN + s2f78 + s2RAUCHEN + s2SCHULE + SES_r + JEMALS_GEARB,
+ family = binomial(link="logit"), data=basis_modell_HEUSCHNUPFEN_amelia1)
> summary(HAY_confounder)
>
> ##############################
> #      Both-Selektion        #
> ##############################
>
> library(MASS)
> HAY_confounder_bothAIC <- stepAIC(HAY_confounder,direction="both",
+ scope = list(upper = HAY_confounder, lower = ~SES_r + f02x))
> summary(HAY_confounder_bothAIC)
```

C.3.2 Schritt 2 - Modelltest

```
> ###############################################################################
> ########## Logit-Modell - Datensatz Amelia1   ###################################
> ###############################################################################
> load("basis_modell_HEUSCHNUPFEN_amelia1.RData")
>
> ##############################
> # Modell 1 #
> ##############################
> HAY_modell1 <- glm(s2CURHAYV ~ PAR_ALL_r + CUR_HAY_r + CURHAYV + f02x + SES_r,
+ family = binomial(link="logit"), data=basis_modell_HEUSCHNUPFEN_amelia1)
> summary(HAY_modell1)
> ##############################
> # Modell 2 #
> ##############################
> HAY_modell2 <- glm(s2CURHAYV ~ PAR_ALL_r + CUR_HAY_r + CURHAYV + f02x + SES_r
+ + STILL_r,
+ family = binomial(link="logit"), data=basis_modell_HEUSCHNUPFEN_amelia1)
> summary(HAY_modell2)
> # Likelihood-Ratio-Test:
> # Modell1 vs. Modell2
> library(lmtest)
> lrtest(HAY_modell1, HAY_modell2)
> ##############################
> # Modell 3 #
> ##############################
> HAY_modell3 <- glm(s2CURHAYV ~ PAR_ALL_r + CUR_HAY_r + CURHAYV + f02x + SES_r
+ + STILL_r + f58x,
+ family = binomial(link="logit"), data=basis_modell_HEUSCHNUPFEN_amelia1)
> summary(HAY_modell3)
> # Likelihood-Ratio-Test:
> # Modell2 vs. Modell3
> library(lmtest)
> lrtest(HAY_modell2, HAY_modell3)
```

C.3.3 Schritt 3 - GAM

```
> ###############################################################################
> ##########               GAM - Datensatz Amelia1             ##################
> ###############################################################################
>
> #### Überprüfen, ob die Expositionen als lineare Terme oder als andere Funktionen
> #### in das Modell eingehen kann
>
> load("basis_modell_HEUSCHNUPFEN_amelia1.RData")
> ########### Exposition kumuliert ###############################################
> library(mgcv)
> gam_kumuliert <- gam(s2CURHAYV ~ PAR_ALL_r + CUR_HAY_r + CURHAYV + f02x + SES_r
+ +  STILL_r + s(HMW_kumuliert) + s(LMW_kumuliert) + s(MIXED_kumuliert) + s(LOWRISK_kumuliert),
+ family = binomial(link="logit"), data=basis_modell_HEUSCHNUPFEN_amelia1)
> summary(gam_kumuliert)
```

```
> pdf("GAM_HAY_kumuliert_geschätzte_Funktionen_amelia1.pdf")
> par(mfrow = c(2,2))
> plot(gam_kumuliert)
> dev.off()
> # nichts quadratisch aufnehmen
>
> ########### Exposition 1.Beruf ##############################################
> library(mgcv)
> gam_ersterberuf <- gam(s2CURHAYV ~ PAR_ALL_r + CUR_HAY_r + CURHAYV + f02x + SES_r
+ +  STILL_r + s(HMW_ersterberuf_gesamt) + s(LMW_ersterberuf_gesamt) +
+ s(MIXED_ersterberuf_gesamt) + s(LOWRISK_ersterberuf_gesamt),
+ family = binomial(link="logit"), data=basis_modell_HEUSCHNUPFEN_amelia1)
> summary(gam_ersterberuf)
> pdf("GAM_HAY_ersterberuf_geschätzte_Funktionen_amelia1.pdf")
> par(mfrow = c(2,2))
> plot(gam_ersterberuf)
> dev.off()
> # nichts quadratisch aufnehmen
>
> ########### Exposition 1.Jahr ###############################################
> library(mgcv)
> gam_erstesjahr <- gam(s2CURHAYV ~ PAR_ALL_r + CUR_HAY_r + CURHAYV + f02x + SES_r
+ +  STILL_r + s(HMW_erstesjahr_gesamt) + s(LMW_erstesjahr_gesamt) +
+ s(MIXED_erstesjahr_gesamt) + s(LOWRISK_erstesjahr_gesamt),
+ family = binomial(link="logit"), data=basis_modell_HEUSCHNUPFEN_amelia1)
> summary(gam_erstesjahr)
> pdf("GAM_HAY_erstesjahr_geschätzte_Funktionen_amelia1.pdf")
> par(mfrow = c(2,2))
> plot(gam_erstesjahr)
> dev.off()
```

C.3.4 Schritt 4 - Expositionsvariablen

```
> ###############################################################################
> ########## Test, ob Expo-Variablen Einfluss haben - Datensatz Amelia 1 #########
> ###############################################################################
>
> load("basis_modell_HEUSCHNUPFEN_amelia1.RData")
> ########################################
> ##### Confounder-Modell definieren    #####
> ########################################
> HAY_confounder_amelia1 <- glm(s2CURHAYV ~ PAR_ALL_r + CUR_HAY_r + CURHAYV + f02x + SES_r
+ + STILL_r ,family = binomial(link="logit"), data=basis_modell_HEUSCHNUPFEN_amelia1)
> summary(HAY_confounder_amelia1)
>
> ########################################
> ##### kumulierte Exposition aufnehmen #####
> ########################################
> HAY_kum_amelia1 <- glm(s2CURHAYV ~ PAR_ALL_r + CUR_HAY_r + CURHAYV + f02x + SES_r
+ + STILL_r + HMW_kumuliert + LMW_kumuliert + MIXED_kumuliert + IRRPEAKS_kumuliert + LOWRISK_kumuliert,
+ family = binomial(link="logit"), data=basis_modell_HEUSCHNUPFEN_amelia1)
> summary(HAY_kum_amelia1)
> # Likelihood-Ratio-Test:
> # Confounder-Modell vs. Confounder-Modell mit allen Expo-Variablen
> library(lmtest)
> lrtest(HAY_confounder_amelia1, HAY_kum_amelia1)
> # keine Expo-Variable hat Einfluss!!!
>
> ########################################
> ##### binäre Exposition aufnehmen     #####
> ########################################
> HAY_binaer_amelia1 <- glm(s2CURHAYV ~ PAR_ALL_r + CUR_HAY_r + CURHAYV + f02x + SES_r
+ + STILL_r + HMW_binaer + LMW_binaer + MIXED_binaer + IRRPEAKS_binaer + LOWRISK_binaer,
+ family = binomial(link="logit"), data=basis_modell_HEUSCHNUPFEN_amelia1)
> summary(HAY_binaer_amelia1)
> # Likelihood-Ratio-Test:
> # Confounder-Modell vs. Confounder-Modell mit allen Expo-Variablen
> library(lmtest)
> lrtest(HAY_confounder_amelia1, HAY_binaer_amelia1)
> # keine Expo-Variable hat Einfluss!!!
>
> ########################################
```

```
> ##### Exposition 1.Jahr aufnehmen     #####
> ##########################################
> HAY_erstesjahr_amelia1 <- glm(s2CURHAYV ~ PAR_ALL_r + CUR_HAY_r + CURHAYV + f02x + SES_r
+ + STILL_r + HMW_erstesjahr_gesamt + LMW_erstesjahr_gesamt + MIXED_erstesjahr_gesamt
+ + IRRPEAKS_erstesjahr_gesamt + LOWRISK_erstesjahr_gesamt,
+ family = binomial(link="logit"), data=basis_modell_HEUSCHNUPFEN_amelia1)
> summary(HAY_erstesjahr_amelia1)
> # Likelihood-Ratio-Test:
> # Confounder-Modell vs. Confounder-Modell mit allen Expo-Variablen
> library(lmtest)
> lrtest(HAY_confounder_amelia1, HAY_erstesjahr_amelia1)
> # keine Expo-Variable hat Einfluss!!!
>
> ##########################################
> ##### Exposition 1.Jahr binär aufnehmen ###
> ##########################################
> HAY_erstesjahrbin_amelia1 <- glm(s2CURHAYV ~ PAR_ALL_r + CUR_HAY_r + CURHAYV + f02x + SES_r
+ + STILL_r + HMW_erstesjahr_binaer + LMW_erstesjahr_binaer + MIXED_erstesjahr_binaer
+ + IRRPEAKS_erstesjahr_binaer + LOWRISK_erstesjahr_binaer,
+ family = binomial(link="logit"), data=basis_modell_HEUSCHNUPFEN_amelia1)
> summary(HAY_erstesjahrbin_amelia1)
> # Likelihood-Ratio-Test:
> # Confounder-Modell vs. Confounder-Modell mit allen Expo-Variablen
> library(lmtest)
> lrtest(HAY_confounder_amelia1, HAY_erstesjahrbin_amelia1)
> # keine Expo-Variable hat Einfluss!!!
>
> ##########################################
> ##### Exposition 1.Beruf aufnehmen    #####
> ##########################################
> HAY_ersterberuf_amelia1 <- glm(s2CURHAYV ~ PAR_ALL_r + CUR_HAY_r + CURHAYV + f02x + SES_r
+ + STILL_r + HMW_ersterberuf_gesamt + LMW_ersterberuf_gesamt + MIXED_ersterberuf_gesamt
+ + IRRPEAKS_ersterberuf_gesamt + LOWRISK_ersterberuf_gesamt,
+ family = binomial(link="logit"), data=basis_modell_HEUSCHNUPFEN_amelia1)
> summary(HAY_ersterberuf_amelia1)
> # Likelihood-Ratio-Test:
> # Confounder-Modell vs. Confounder-Modell mit allen Expo-Variablen
> library(lmtest)
> lrtest(HAY_confounder_amelia1, HAY_ersterberuf_amelia1)
> # keine Expo-Variable hat Einfluss!!!
>
> ##########################################
> ##### Exposition 1.Beruf binär aufnehmen ##
> ##########################################
> HAY_ersterberufbin_amelia1 <- glm(s2CURHAYV ~ PAR_ALL_r + CUR_HAY_r + CURHAYV + f02x + SES_r
+ + STILL_r + HMW_ersterberuf_binaer + LMW_ersterberuf_binaer + MIXED_ersterberuf_binaer
+ + IRRPEAKS_ersterberuf_binaer + LOWRISK_ersterberuf_binaer,
+ family = binomial(link="logit"), data=basis_modell_HEUSCHNUPFEN_amelia1)
> summary(HAY_ersterberufbin_amelia1)
> # Likelihood-Ratio-Test:
> # Confounder-Modell vs. Confounder-Modell mit allen Expo-Variablen
> library(lmtest)
> lrtest(HAY_confounder_amelia1, HAY_ersterberufbin_amelia1)
> # keine Expo-Variable hat Einfluss!!!
```

C.3.5 Schritt 5 - Bestes Modell

```
> ##############################################################################
> ########## Bestes Modell - Datensatz Amelia 1 ################################
> ##############################################################################
>
> load("basis_modell_HEUSCHNUPFEN_amelia1.RData")
> ### bestes Modell
> HAY_bestes_modell_amelia1 <- glm(s2CURHAYV ~ PAR_ALL_r + CUR_HAY_r + CURHAYV + f02x + SES_r
+ + STILL_r,
+ family = binomial(link="logit"), data=basis_modell_HEUSCHNUPFEN_amelia1)
> summary(HAY_bestes_modell_amelia1)
> # Schätzer exponieren
> exp(coefficients(HAY_bestes_modell_amelia1))
> ### Konfidenzintervall
> KI <- exp(confint(HAY_bestes_modell_amelia1))
> pdf("KI_heuschnupfen_amelia1.pdf")
```

```
> plot( c(0, 40) , c(0, (nrow(KI)-1)), type="n", xlab="",ylab="",
+ main="Logit-Modell - Konfidenzintervalle der Schätzer",sub="Amelia1")
> # "Pfeil" in beide Richtungen zeichnen
> axis(1,at=c(0,1,2,3,4,5,10,20,30,40,50))
> arrows(KI[1,1], 6, KI[1,2], 6, angle=90, length=0.1, code=3, lwd=2)
> text( 35, 6, "Intercept",adj=c(0,0))
> arrows(KI[2,1], 5, KI[2,2], 5, angle=90, length=0.1, code=3, lwd=2)
> text( 35, 5, "PAR_ALL_r",adj=c(0,0))
> arrows(KI[3,1], 4, KI[3,2], 4, angle=90, length=0.1, code=3, lwd=2)
> text( 35, 4, "CUR_HAY_r",adj=c(0,0))
> arrows(KI[4,1], 3, KI[4,2], 3, angle=90, length=0.1, code=3, lwd=2)
> text( 35, 3, "CURHAYV",adj=c(0,0))
> arrows(KI[5,1], 2, KI[5,2], 2, angle=90, length=0.1, code=3, lwd=2)
> text( 35, 2, "f02x",adj=c(0,0))
> arrows(KI[6,1], 1, KI[6,2], 1, angle=90, length=0.1, code=3, lwd=2)
> text( 35, 1, "SES_r",adj=c(0,0))
> arrows(KI[7,1], 0, KI[7,2], 0, angle=90, length=0.1, code=3, lwd=2)
> text( 35, 0, "STILL_r",adj=c(0,0))
> abline(v=1,lty=2)
> dev.off()
> ### ROC-Kurve
> library(Epi)
> attach(basis_modell_HEUSCHNUPFEN_amelia1)
> pdf("roc_heuschnupfen_amelia1.pdf")
> roc_am1 <- ROC(form = s2CURHAYV ~ PAR_ALL_r + CUR_HAY_r + CURHAYV + f02x + SES_r
+ +  STILL_r ,
+ plot = "ROC", # nur ROC Kurve soll gezeichnet werden
+ PV = FALSE, # Sensitivität, Spezifität, Prädiktive Werte am optimalen Cutpoint
+             # angeben lassen
+ MX = FALSE, # Optimaler Cutpoint (wo Sensitivität und Spezifität optimal)
+            # angeben lassen
+ AUC = TRUE, # Area under Curve zeichnen lassen
+ lwd = 2,
+ MI=FALSE)  # Model summary des logist. Modells mit reinschreiben lassen
> dev.off()
> detach(basis_modell_HEUSCHNUPFEN_amelia1)
```

C.3.6 Schritt 6 - Schätzer kombinieren

```
> ###############################################################################
> ########## Bestes Modell - Datensatz Amelia 1 ##################################
> ###############################################################################
>
> load("basis_modell_HEUSCHNUPFEN_amelia1.RData")
> ### bestes Modell
> HAY_bestes_modell_amelia1 <- glm(s2CURHAYV ~ PAR_ALL_r + CUR_HAY_r + CURHAYV + f02x + SES_r
+ + STILL_r,
+ family = binomial(link="logit"), data=basis_modell_HEUSCHNUPFEN_amelia1)
> summary(HAY_bestes_modell_amelia1)
> ###############################################################################
> ########## Bestes Modell - Datensatz Amelia 2 ##################################
> ###############################################################################
>
> load("basis_modell_HEUSCHNUPFEN_amelia2.RData")
> ### bestes Modell
> HAY_bestes_modell_amelia2 <- glm(s2CURHAYV ~ PAR_ALL_r + CUR_HAY_r + CURHAYV + f02x + SES_r
+ + STILL_r,
+ family = binomial(link="logit"), data=basis_modell_HEUSCHNUPFEN_amelia2)
> summary(HAY_bestes_modell_amelia2)
> ###############################################################################
> ########## Bestes Modell - Datensatz Amelia 3 ##################################
> ###############################################################################
>
> load("basis_modell_HEUSCHNUPFEN_amelia3.RData")
> ### bestes Modell
> HAY_bestes_modell_amelia3 <- glm(s2CURHAYV ~ PAR_ALL_r + CUR_HAY_r + CURHAYV + f02x + SES_r
+ + STILL_r,
+ family = binomial(link="logit"), data=basis_modell_HEUSCHNUPFEN_amelia3)
> summary(HAY_bestes_modell_amelia3)
> ###############################################################################
> ########## Bestes Modell - Datensatz EmpVert 1 #################################
> ###############################################################################
```

```
>
> load("basis_modell_HEUSCHNUPFEN_empVert1.RData")
> ### bestes Modell
> HAY_bestes_modell_empVert1 <- glm(s2CURHAYV ~ PAR_ALL_r + CUR_HAY_r + CURHAYV + f02x + SES_r
+ + STILL_r,
+ family = binomial(link="logit"), data=basis_modell_HEUSCHNUPFEN_empVert1)
> summary(HAY_bestes_modell_empVert1)
> ##############################################################################
> ########## Bestes Modell - Datensatz EmpVert 2 ################################
> ##############################################################################
>
> load("basis_modell_HEUSCHNUPFEN_empVert2.RData")
> ### bestes Modell
> HAY_bestes_modell_empVert2 <- glm(s2CURHAYV ~ PAR_ALL_r + CUR_HAY_r + CURHAYV + f02x + SES_r
+ + STILL_r,
+ family = binomial(link="logit"), data=basis_modell_HEUSCHNUPFEN_empVert2)
> summary(HAY_bestes_modell_empVert2)
> ##############################################################################
> #########               Schätzer kombinieren                         #########
> ##############################################################################
>
> # Schätzer aus den einzelnen Datensätzen
> ca1 <- coefficients(HAY_bestes_modell_amelia1)
> ca2 <- coefficients(HAY_bestes_modell_amelia2)
> ca3 <- coefficients(HAY_bestes_modell_amelia3)
> ce1 <- coefficients(HAY_bestes_modell_empVert1)
> ce2 <- coefficients(HAY_bestes_modell_empVert2)
> # Parameterschätzer: Intercept
> intercept_quer <- 1/5 * (ca1[1]+ca2[1]+ca3[1]+ce1[1]+ce2[1])
> # Parameterschätzer: PAR_ALL_r
> PAR_ALL_r_quer <- 1/5 * (ca1[2]+ca2[2]+ca3[2]+ce1[2]+ce2[2])
> # Parameterschätzer: CUR_HAY_r
> CUR_HAY_r_quer <- 1/5 * (ca1[3]+ca2[3]+ca3[3]+ce1[3]+ce2[3])
> # Parameterschätzer: CURHAYV
> CURHAYV_quer <- 1/5 * (ca1[4]+ca2[4]+ca3[4]+ce1[4]+ce2[4])
> # Parameterschätzer: Geschlecht (f02x)
> f02x_quer <- 1/5 * (ca1[5]+ca2[5]+ca3[5]+ce1[5]+ce2[5])
> # Parameterschätzer: SES_r
> SES_r_quer <- 1/5 * (ca1[6]+ca2[6]+ca3[6]+ce1[6]+ce2[6])
> # Parameterschätzer: STILL_r
> STILL_r_quer <- 1/5 * (ca1[7]+ca2[7]+ca3[7]+ce1[7]+ce2[7])
> # Varianz-Kovarianz-Matrizen der Schätzer
> vcov_a1 <- vcov(HAY_bestes_modell_amelia1)
> vcov_a2 <- vcov(HAY_bestes_modell_amelia2)
> vcov_a3 <- vcov(HAY_bestes_modell_amelia3)
> vcov_e1 <- vcov(HAY_bestes_modell_empVert1)
> vcov_e2 <- vcov(HAY_bestes_modell_empVert2)
>
> # Varianz innerhalb jedes Datensatzes ist das arithmetische Mittel der
> # geschaetzten Varianzen:
> # Varianz innerhalb des Datensatzes: Intercept
> intercept_var_innerhalb <- 1/5 * (vcov_a1[1,1]+vcov_a2[1,1]+vcov_a3[1,1]
+ +vcov_e1[1,1]+vcov_e2[1,1])
> # Varianz innerhalb des Datensatzes: PAR_ALL_r
> PAR_ALL_r_var_innerhalb <- 1/5 * (vcov_a1[2,2]+vcov_a2[2,2]+vcov_a3[2,2]
+ +vcov_e1[2,2]+vcov_e2[2,2])
> # Varianz innerhalb des Datensatzes: CUR_HAY_r
> CUR_HAY_r_var_innerhalb <- 1/5 * (vcov_a1[3,3]+vcov_a2[3,3]+vcov_a3[3,3]
+ +vcov_e1[3,3]+vcov_e2[3,3])
> # Varianz innerhalb des Datensatzes: CURHAYV
> CURHAYV_var_innerhalb <- 1/5 * (vcov_a1[4,4]+vcov_a2[4,4]+vcov_a3[4,4]
+ +vcov_e1[4,4]+vcov_e2[4,4])
> # Varianz innerhalb des Datensatzes: Geschlecht (f02x)
> f02x_var_innerhalb <- 1/5 * (vcov_a1[5,5]+vcov_a2[5,5]+vcov_a3[5,5]
+ +vcov_e1[5,5]+vcov_e2[5,5])
> # Varianz innerhalb des Datensatzes: SES_r
> SES_r_var_innerhalb <- 1/5 * (vcov_a1[6,6]+vcov_a2[6,6]+vcov_a3[6,6]
+ +vcov_e1[6,6]+vcov_e2[6,6])
> # Varianz innerhalb des Datensatzes: STILL_r
> STILL_r_var_innerhalb <- 1/5 * (vcov_a1[7,7]+vcov_a2[7,7]+vcov_a3[7,7]
+ +vcov_e1[7,7]+vcov_e2[7,7])
>
> # Die Varianz zwischen den Datensaetzen ist die Stichprobenvarianz der
```

```
> # Schaetzer selbst:
> # Varianz zwischen den Datensaetzen: Intercept
> intercept_var_zwischen <- 1/4 * ((ca1[1]-intercept_quer)^2
+ +(ca2[1]-intercept_quer)^2+(ca3[1]-intercept_quer)^2
+ +(ce1[1]-intercept_quer)^2+(ce2[1]-intercept_quer)^2)
> # Varianz zwischen den Datensaetzen: PAR_ALL_r
> PAR_ALL_r_var_zwischen <- 1/4 * ((ca1[2]-PAR_ALL_r_quer)^2
+ +(ca2[2]-PAR_ALL_r_quer)^2+(ca3[2]-PAR_ALL_r_quer)^2
+ +(ce1[2]-PAR_ALL_r_quer)^2+(ce2[2]-PAR_ALL_r_quer)^2)
> # Varianz zwischen den Datensaetzen: CUR_HAY_r
> CUR_HAY_r_var_zwischen <- 1/4 * ((ca1[3]-CUR_HAY_r_quer)^2
+ +(ca2[3]-CUR_HAY_r_quer)^2+(ca3[3]-CUR_HAY_r_quer)^2
+ +(ce1[3]-CUR_HAY_r_quer)^2+(ce2[3]-CUR_HAY_r_quer)^2)
> # Varianz zwischen den Datensaetzen: CURHAYV
> CURHAYV_var_zwischen <- 1/4 * ((ca1[4]-CURHAYV_quer)^2
+ +(ca2[4]-CURHAYV_quer)^2+(ca3[4]-CURHAYV_quer)^2
+ +(ce1[4]-CURHAYV_quer)^2+(ce2[4]-CURHAYV_quer)^2)
> # Varianz zwischen den Datensaetzen: Geschlecht (f02x)
> f02x_var_zwischen <- 1/4 * ((ca1[5]-f02x_quer)^2
+ +(ca2[5]-f02x_quer)^2+(ca3[5]-f02x_quer)^2
+ +(ce1[5]-f02x_quer)^2+(ce2[5]-f02x_quer)^2)
> # Varianz zwischen den Datensaetzen: SES_r
> SES_r_var_zwischen <- 1/4 * ((ca1[6]-SES_r_quer)^2
+ +(ca2[6]-SES_r_quer)^2+(ca3[6]-SES_r_quer)^2
+ +(ce1[6]-SES_r_quer)^2+(ce2[6]-SES_r_quer)^2)
> # Varianz zwischen den Datensaetzen: STILL_r
> STILL_r_var_zwischen <- 1/4 * ((ca1[7]-STILL_r_quer)^2
+ +(ca2[7]-STILL_r_quer)^2+(ca3[7]-STILL_r_quer)^2
+ +(ce1[7]-STILL_r_quer)^2+(ce2[7]-STILL_r_quer)^2)
>
> # Die Gesamtvarianz T entspricht der Summe der beiden Komponenten mit einem
> # zusaetzlichen Korrekturfaktor fuer den Simulationsfehler in Q_quer
> # Gesamtvarianz: Intercept
> intercept_var_gesamt <- (intercept_var_innerhalb
+ +(1+(1/5))*intercept_var_zwischen)
> # Gesamtvarianz: PAR_ALL_r
> PAR_ALL_r_var_gesamt <- (PAR_ALL_r_var_innerhalb
+ +(1+(1/5))*PAR_ALL_r_var_zwischen)
> # Gesamtvarianz: CUR_HAY_r
> CUR_HAY_r_var_gesamt <- (CUR_HAY_r_var_innerhalb+(1+(1/5))*CUR_HAY_r_var_zwischen)
> # Gesamtvarianz: CURHAYV
> CURHAYV_var_gesamt <- (CURHAYV_var_innerhalb+(1+(1/5))*CURHAYV_var_zwischen)
> # Gesamtvarianz: Geschlecht (f02x)
> f02x_var_gesamt <- (f02x_var_innerhalb+(1+(1/5))*f02x_var_zwischen)
> # Gesamtvarianz: SES_r
> SES_r_var_gesamt <- (SES_r_var_innerhalb+(1+(1/5))*SES_r_var_zwischen)
> # Gesamtvarianz: STILL_r
> STILL_r_var_gesamt <- (STILL_r_var_innerhalb+(1+(1/5))*STILL_r_var_zwischen)
>
> # Standardabweichungen der Schätzer: sqrt(Varianz)
> # Standardabweichung: Intercept
> intercept_stdabw <- sqrt(intercept_var_gesamt)
> # Standardabweichung: PAR_ALL_r
> PAR_ALL_r_stdabw <- sqrt(PAR_ALL_r_var_gesamt)
> # Standardabweichung: CUR_HAY_r
> CUR_HAY_r_stdabw <- sqrt(CUR_HAY_r_var_gesamt)
> # Standardabweichung: CURHAYV
> CURHAYV_stdabw <- sqrt(CURHAYV_var_gesamt)
> # Standardabweichung: Geschlecht (f02x)
> f02x_stdabw <- sqrt(f02x_var_gesamt)
> # Standardabweichung: SES_r
> SES_r_stdabw <- sqrt(SES_r_var_gesamt)
> # Standardabweichung: STILL_r
> STILL_r_stdabw <- sqrt(STILL_r_var_gesamt)
>
> ##############################################################################
> #########    (kombinierte) Konfidenzintervalle berechnen und zeichen   #########
> ##############################################################################
>
> KI_u_intercept <- exp(intercept_quer - 1.96 * intercept_stdabw)
> KI_o_intercept <- exp(intercept_quer + 1.96 * intercept_stdabw)
> KI_u_PAR_ALL_r <- exp(PAR_ALL_r_quer - 1.96 * PAR_ALL_r_stdabw)
> KI_o_PAR_ALL_r <- exp(PAR_ALL_r_quer + 1.96 * PAR_ALL_r_stdabw)
```

```
> KI_u_CUR_HAY_r <- exp(CUR_HAY_r_quer - 1.96 * CUR_HAY_r_stdabw)
> KI_o_CUR_HAY_r <- exp(CUR_HAY_r_quer + 1.96 * CUR_HAY_r_stdabw)
> KI_u_CURHAYV <- exp(CURHAYV_quer - 1.96 * CURHAYV_stdabw)
> KI_o_CURHAYV <- exp(CURHAYV_quer + 1.96 * CURHAYV_stdabw)
> KI_u_f02x <- exp(f02x_quer - 1.96 * f02x_stdabw)
> KI_o_f02x <- exp(f02x_quer + 1.96 * f02x_stdabw)
> KI_u_SES_r <- exp(SES_r_quer - 1.96 * SES_r_stdabw)
> KI_o_SES_r <- exp(SES_r_quer + 1.96 * SES_r_stdabw)
> KI_u_STILL_r <- exp(STILL_r_quer - 1.96 * STILL_r_stdabw)
> KI_o_STILL_r <- exp(STILL_r_quer + 1.96 * STILL_r_stdabw)
>
> pdf("KI_heuschnupfen_kombiniert.pdf")
> plot( c(0, 61) , c(0, 6), type="n", xlab="",ylab="",
+ main="Konfidenzintervalle der Odds-Ratios",sub="Logit-Modell für Allergische Rhinitis")
> # "Pfeil" in beide Richtungen zeichnen
> axis(1,at=c(0,1,2,3,4,5,10,20,30,40,50))
> arrows(KI_u_intercept, 6, KI_o_intercept, 6, angle=90, length=0.1, code=3, lwd=2)
> text( 35, 6, "Intercept",adj=c(0,0))
> arrows(KI_u_PAR_ALL_r, 5, KI_o_PAR_ALL_r, 5, angle=90, length=0.1, code=3, lwd=2)
> text( 35, 5, "Atopie der Eltern",adj=c(0,0))
> arrows(KI_u_CUR_HAY_r, 4, KI_o_CUR_HAY_r, 4, angle=90, length=0.1, code=3, lwd=2)
> text( 35, 4, "Allergische Rhinitis (ISAAC II)",adj=c(0,0))
> arrows(KI_u_CURHAYV, 3, KI_o_CURHAYV, 3, angle=90, length=0.1, code=3, lwd=2)
> text( 35, 3, "Allergische Rhinitis (SOLAR)",adj=c(0,0))
> arrows(KI_u_f02x, 2, KI_o_f02x, 2, angle=90, length=0.1, code=3, lwd=2)
> text( 35, 2, "Geschlecht",adj=c(0,0))
> arrows(KI_u_SES_r, 1, KI_o_SES_r, 1, angle=90, length=0.1, code=3, lwd=2)
> text( 35, 1, "Sozioökonomischer Status",adj=c(0,0))
> arrows(KI_u_STILL_r, 0, KI_o_STILL_r, 0, angle=90, length=0.1, code=3, lwd=2)
> text( 35, 0, "Als Säugling gestillt",adj=c(0,0))
> abline(v=1,lty=2)
> dev.off()
```

C.4 Simulation

C.4.1 Schritt 1 - Werte künstlich löschen

```
> ##############################################################################
> ######################## Fehlendmuster erstellen ##############################
> ##############################################################################
>
> # 25 mal aus dem Datensatz "berufsdaten_vollständig", der die vollständigen
> # Berufsdaten enthält Werte zufällig löschen
>
> load("berufsdaten_vollstaendig.RData")
> nrow(berufsdaten_vollstaendig) # hat 10940 Zeilen, da 1094 Probanden mit
> # vollständigen Berufsangaben - pro Proband 10 Zeilen für max. 10 Berufe
>
> # Index einführen damit man eindeutig eine Kombi von Proband u.Beruf ziehen kann
> berufsdaten_vollstaendig$index <- 1:10940
> berufsdaten_vollstaendig_plus_index <- berufsdaten_vollstaendig
> save(berufsdaten_vollstaendig_plus_index,
+ file="berufsdaten_vollstaendig_plus_index.RData")
> # Nur aus den Probanden ziehen die auch wirklich Angaben gemacht haben
> angaben_vorhanden <- subset(berufsdaten_vollstaendig_plus_index,
+ !is.na(ANF_JAHR) & !is.na(ANF_MONAT) & !is.na(END_JAHRx) & !is.na(END_MONATx)
+ & !is.na(WST) & ISCO!=94 & ISCO!=95 & ISCO!=97 & ISCO!=98)
> nrow(angaben_vorhanden)   # sind 1447 Zeilen
>
> ##############################################################################
> ####       FUNKTION UM WERTE AUS BERUFSDATEN KÜNSTLICH ZU LÖSCHEN          ####
> ##############################################################################
>
> # Argumente der Funktion: daten (Datensatz der vollständige Berufsdaten enthält)
> # und startwert (damit zufällig und nachvollziehbar gezogen wird)
>
> Werte_kuenstl_loeschen <- function(daten, startwert){
+
+ # Es sind insgesamt 101 Zeilen bei denen was fehlen soll im Datensatz
+ # => 101 Indizes aus Datensatz angaben_vorhanden ohne Zurücklegen ziehen und in
```

```
+ # einen Vektor (auswahl) speichern
+ set.seed(startwert)
+ auswahl <- as.vector(sample(angaben_vorhanden$index, 101, replace=FALSE))
+
+ # Indikator, der angibt ob Werte künstlich gelöscht wurden
+ daten$kuenstl_geloescht <- 0
+
+ # Bei den ersten 12 Indizes aus dem Vektor sollen alle Angaben bis auf der
+ # ISCO-Code fehlen => alles bis auf ISCO-Code löschen
+ for(i in 1:nrow(daten)){
+ if ((auswahl[1] == daten$index[i]) |
+ (auswahl[2] == daten$index[i]) |
+ (auswahl[3] == daten$index[i]) |
+ (auswahl[4] == daten$index[i]) |
+ (auswahl[5] == daten$index[i]) |
+ (auswahl[6] == daten$index[i]) |
+ (auswahl[7] == daten$index[i]) |
+ (auswahl[8] == daten$index[i]) |
+ (auswahl[9] == daten$index[i]) |
+ (auswahl[10] == daten$index[i]) |
+ (auswahl[11] == daten$index[i]) |
+ (auswahl[12] == daten$index[i])
+ ){
+ daten$ANF_JAHR[i] <- NA
+ daten$ANF_MONAT[i] <- NA
+ daten$END_JAHRx[i] <- NA
+ daten$END_MONATx[i] <- NA
+ daten$WST[i] <- NA
+ daten$kuenstl_geloescht[i] <- 1
+ }
+ }
+
+ # Bei den nächsten 25 Indizes aus dem Vektor sollen nur die WST gelöscht werden
+ for(i in 1:nrow(daten)){
+ if ((auswahl[13] == daten$index[i]) |
+ (auswahl[14] == daten$index[i]) |
+ (auswahl[15] == daten$index[i]) |
+ (auswahl[16] == daten$index[i]) |
+ (auswahl[17] == daten$index[i]) |
+ (auswahl[18] == daten$index[i]) |
+ (auswahl[19] == daten$index[i]) |
+ (auswahl[20] == daten$index[i]) |
+ (auswahl[21] == daten$index[i]) |
+ (auswahl[22] == daten$index[i]) |
+ (auswahl[23] == daten$index[i]) |
+ (auswahl[24] == daten$index[i]) |
+ (auswahl[25] == daten$index[i]) |
+ (auswahl[26] == daten$index[i]) |
+ (auswahl[27] == daten$index[i]) |
+ (auswahl[28] == daten$index[i]) |
+ (auswahl[29] == daten$index[i]) |
+ (auswahl[30] == daten$index[i]) |
+ (auswahl[31] == daten$index[i]) |
+ (auswahl[32] == daten$index[i]) |
+ (auswahl[33] == daten$index[i]) |
+ (auswahl[34] == daten$index[i]) |
+ (auswahl[35] == daten$index[i]) |
+ (auswahl[36] == daten$index[i]) |
+ (auswahl[37] == daten$index[i])
+ ){
+ daten$WST[i] <- NA
+ daten$kuenstl_geloescht[i] <- 1
+ }
+ }
+
+ # Bei den nächsten 13 Indizes aus dem Vektor sollen WST, Zeitangaben zum Ende
+ # des Berufs gelöscht werden
+ for(i in 1:nrow(daten)){
+ if ((auswahl[38] == daten$index[i]) |
+ (auswahl[39] == daten$index[i]) |
+ (auswahl[40] == daten$index[i]) |
+ (auswahl[41] == daten$index[i]) |
+ (auswahl[42] == daten$index[i]) |
```

```
+ (auswahl[43] == daten$index[i]) |
+ (auswahl[44] == daten$index[i]) |
+ (auswahl[45] == daten$index[i]) |
+ (auswahl[46] == daten$index[i]) |
+ (auswahl[47] == daten$index[i]) |
+ (auswahl[48] == daten$index[i]) |
+ (auswahl[49] == daten$index[i]) |
+ (auswahl[50] == daten$index[i])
+ ){
+ daten$END_JAHRx[i] <- NA
+ daten$END_MONATx[i] <- NA
+ daten$WST[i] <- NA
+ daten$kuenstl_geloescht[i] <- 1
+ }
+ }
+
+ # Bei den nächsten 13 Indizes aus dem Vektor sollen Zeitangaben zum Anfang und
+ # Ende des Berufs gelöscht werden
+ for(i in 1:nrow(daten)){
+ if ((auswahl[51] == daten$index[i]) |
+ (auswahl[52] == daten$index[i]) |
+ (auswahl[53] == daten$index[i]) |
+ (auswahl[54] == daten$index[i]) |
+ (auswahl[55] == daten$index[i]) |
+ (auswahl[56] == daten$index[i]) |
+ (auswahl[57] == daten$index[i]) |
+ (auswahl[58] == daten$index[i]) |
+ (auswahl[59] == daten$index[i]) |
+ (auswahl[60] == daten$index[i]) |
+ (auswahl[61] == daten$index[i]) |
+ (auswahl[62] == daten$index[i]) |
+ (auswahl[63] == daten$index[i])
+ ){
+ daten$ANF_JAHR[i] <- NA
+ daten$ANF_MONAT[i] <- NA
+ daten$END_JAHRx[i] <- NA
+ daten$END_MONATx[i] <- NA
+ daten$kuenstl_geloescht[i] <- 1
+ }
+ }
+
+ # Bei den nächsten 16 Indizes aus dem Vektor sollen Anfangsmonat und Endmonat
+ # gelöscht werden
+ for(i in 1:nrow(daten)){
+ if ((auswahl[64] == daten$index[i]) |
+ (auswahl[65] == daten$index[i]) |
+ (auswahl[66] == daten$index[i]) |
+ (auswahl[67] == daten$index[i]) |
+ (auswahl[68] == daten$index[i]) |
+ (auswahl[69] == daten$index[i]) |
+ (auswahl[70] == daten$index[i]) |
+ (auswahl[71] == daten$index[i]) |
+ (auswahl[72] == daten$index[i]) |
+ (auswahl[73] == daten$index[i]) |
+ (auswahl[74] == daten$index[i]) |
+ (auswahl[75] == daten$index[i]) |
+ (auswahl[76] == daten$index[i]) |
+ (auswahl[77] == daten$index[i]) |
+ (auswahl[78] == daten$index[i]) |
+ (auswahl[79] == daten$index[i])
+ ){
+ daten$ANF_MONAT[i] <- NA
+ daten$END_MONATx[i] <- NA
+ daten$kuenstl_geloescht[i] <- 1
+ }
+ }
+
+ # Bei den nächsten 14 Indizes aus dem Vektor sollen alle Zeitangaben bis auf das
+ # Anfangsjahr gelöscht werden
+ for(i in 1:nrow(daten)){
+ if ((auswahl[80] == daten$index[i]) |
+ (auswahl[81] == daten$index[i]) |
+ (auswahl[82] == daten$index[i]) |
```

```
+ (auswahl[83] == daten$index[i]) |
+ (auswahl[84] == daten$index[i]) |
+ (auswahl[85] == daten$index[i]) |
+ (auswahl[86] == daten$index[i]) |
+ (auswahl[87] == daten$index[i]) |
+ (auswahl[88] == daten$index[i]) |
+ (auswahl[89] == daten$index[i]) |
+ (auswahl[90] == daten$index[i]) |
+ (auswahl[91] == daten$index[i]) |
+ (auswahl[92] == daten$index[i]) |
+ (auswahl[93] == daten$index[i])
+ ){
+ daten$ANF_MONAT[i] <- NA
+ daten$END_JAHRx[i] <- NA
+ daten$END_MONATx[i] <- NA
+ daten$kuenstl_geloescht[i] <- 1
+ }
+ }
+
+ # Bei den nächsten 1 Indizes aus dem Vektor sollen Anfangsjahr und Endjahr
+ # gelöscht werden
+ for(i in 1:nrow(daten)){
+ if ((auswahl[94] == daten$index[i])
+ ){
+ daten$ANF_JAHR[i] <- NA
+ daten$END_JAHRx[i] <- NA
+ daten$kuenstl_geloescht[i] <- 1
+ }
+ }
+
+ # Beim nächsten Index aus dem Vektor soll nur Anfangsmonat gelöscht werden
+ for(i in 1:nrow(daten)){
+ if ((auswahl[95] == daten$index[i])
+ ){
+ daten$ANF_MONAT[i] <- NA
+ daten$kuenstl_geloescht[i] <- 1
+ }
+ }
+
+ # Bei den nächsten 3 Indizes aus dem Vektor sollen WST und Zeitangaben bis auf
+ # Anfangsjahr gelöscht werden
+ for(i in 1:nrow(daten)){
+ if ((auswahl[96] == daten$index[i]) |
+ (auswahl[97] == daten$index[i]) |
+ (auswahl[98] == daten$index[i])
+ ){
+ daten$ANF_MONAT[i] <- NA
+ daten$END_JAHRx[i] <- NA
+ daten$END_MONATx[i] <- NA
+ daten$WST[i] <- NA
+ daten$kuenstl_geloescht[i] <- 1
+ }
+ }
+
+ # Beim nächsten Index aus dem Vektor sollen Anfangsmonat und WST gelöscht werden
+ for(i in 1:nrow(daten)){
+ if ((auswahl[99] == daten$index[i])
+ ){
+ daten$ANF_MONAT[i] <- NA
+ daten$WST[i] <- NA
+ daten$kuenstl_geloescht[i] <- 1
+ }
+ }
+
+ # Beim nächsten Index aus dem Vektor sollen Zeitangaben bis auf Anfangsmonat
+ # gelöscht werden
+ for(i in 1:nrow(daten)){
+ if ((auswahl[100] == daten$index[i])
+ ){
+ daten$ANF_JAHR[i] <- NA
+ daten$END_JAHRx[i] <- NA
+ daten$END_MONATx[i] <- NA
+ daten$kuenstl_geloescht[i] <- 1
```

```
+ }
+ }
+
+ # Beim nächsten Index aus dem Vektor soll nur Endjahr gelöscht werden
+ for(i in 1:nrow(daten)){
+ if ((auswahl[101] == daten$index[i])
+ ){
+ daten$END_JAHRx[i] <- NA
+ daten$kuenstl_geloescht[i] <- 1
+ }
+ }
+
+ # daten als Daten_kuenstl_geloescht speichern
+ Daten_kuenstl_geloescht <- daten
+ # Datensatz zurückgeben lassen
+ return(Daten_kuenstl_geloescht)
+ }
> ##############################################################################
> ####     Funktion auf Datensatz ausführen - Hier nur beispielhaft          ####
> ##############################################################################
>
> # 25 Startwerte
> startwert1 <- c(2412,8,924,184,12,7435,214,12318,4,91328,7329,1275,34,163,93189,
+ 3275,42,312,983,721,14,987,213,2148,9481)
> # Funktion 25 mal ausführen => 25 mal Werte löschen
> Werte_kuenstl_geloescht1 <- Werte_kuenstl_loeschen(
+ berufsdaten_vollstaendig_plus_index, startwert=startwert1[1])
> # usw.
>
> # Alle in eine Liste abspeichern
> Datensaetze_kuenstl_geloescht <- vector("list",25)
> Datensaetze_kuenstl_geloescht[[1]] <- Werte_kuenstl_geloescht1
> # usw.
> # Liste die Datensätze mit künstlich gelöschten Werten enthält abspeichern
> save(Datensaetze_kuenstl_geloescht, file="Datensaetze_kuenstl_geloescht.RData")
```

C.4.2 Schritt 2 - Imputation der fehlenden Werte in den Tätigkeitsdaten

Imputation der fehlenden Tätigkeit

Hier ist auch noch der Teil zur Imputation von Expositionen enthalten, der so in der Bachelorarbeit nicht benötigt wurden aber evtl. mal hilfreich sein könnte.
Hat ein Proband keine Angaben zur Tätigkeit sondern nur zu den Zeitangaben und der Anzahl der Wochenstunden gemacht, so konnte die entsprechende Tätigkeit nicht mit Hilfe eines ISCO-Codes kodiert werden. Aufgrund des fehlenden ISCO-Codes konnte dann auch keine asthmaspezifische Exposition aus der Job-Exposure-Matrix zugeordnet werden. Die Exposition fehlt also bei fehlendem ISCO-Code auch und muss imputiert werden. Die Exposition besteht aus 22 binären Variablen (Tier-, Latex-, Mehlexposition usw.), die jeweils mit 0 (keine Exposition) und 1 (Exposition) kodiert sind. Die Expositionen in diesen einzelnen binären Variablen sind allerdings nicht frei kombinierbar, sondern treten in bestimmten Mustern auf. Zum Beispiel gibt es bei Tierärzten die Kombination aus Tierexposition und Latexexposition. Die Kombination aus Tierexposition und Mehlexposition tritt dagegen nicht auf. Darum wurden für die Imputation der Exposition zunächst die verschiedenen Expositionsmuster, die auch wirklich in den beobachteten Daten aufgetreten sind, ermittelt und deren Auftrittshäufigkeiten bestimmt. Somit kann anschließend für die Imputation der Exposition bei fehlendem ISCO-Code ein Expositionsmuster aus der empirischen Verteilung gezogen und imputiert werden.

```
> ##############################################################################
> ####           FUNKTION FÜR DIE IMPUTATION DER BERUFSDATEN                ####
> ##############################################################################
>
> # Funktion für die Imputation der Berufsdaten: Als Argumente werden eingegeben:
> # datensatz (enthält fehlende Werte in den Berufsdaten, Fragebogendaten vollst.)
> # startwert (um immer eine andere zufällige Ziehung der Werte für die Imputation
> # zu erhalten und es nachvollziehbar zu machen)
>
> Imputation_Berufsdaten <- function(datensatz, startwert){
+
+ # Subset für SOLAR I:
+ datensatz_s1 <- subset(datensatz, STUDIE == 1)
+ # Subset für SOLAR II:
+ datensatz_s2 <- subset(datensatz, STUDIE == 2)
+
+ ##############################################################################
```

```
+ ####                  Wochenstunden (WST) imputieren                       ####
+ ###############################################################################
+
+ # Indikator, der angibt ob die Wochenstunden imputiert wurden anlegen
+ datensatz$IMP_WST <- 0
+
+ ############################## SOLAR I #######################################
+ print("**************Imputation der Wochenstunden in SOLAR I***************")
+ a1 <- subset(datensatz_s1, is.na(WST) & kuenstl_geloescht == 1)
+ # hier nur die Zeilen drin in denen die WST künstlich gelöscht wurden
+
+ # Datensatz anlegen, in den jeweils index der Zeile und imputierte WST
+ # geschrieben werden (1.Spalte: index, 2.Spalte: wochenstunden)
+ imput_werte1 <- data.frame(index=numeric(nrow(a1)),
+ wochenstunden = numeric(nrow(a1)))
+
+ # j auf 1 setzen
+ j <- 1
+
+ # Startwert setzen (der der Funktion als Argument übergeben wurde)
+ set.seed(startwert)
+
+ for (i in 1: nrow(a1)){
+ print("Nächste Stelle i")
+ # Geschlecht,ISCO und Index an der i-ten Stelle aus subset betrachten
+ isco <- a1$ISCO[i]
+ print("ISCO an der Stelle i")
+ print(isco)
+ geschlecht <- a1$f02x[i]
+ print("Geschlecht an der Stelle i")
+ print(geschlecht)
+ index <- a1$index[i]
+ print("Index an der Stelle i")
+ print(index)
+ # aus großen Datensatz alle mit gleichem ISCO und Geschlecht ziehen, durch
+ # IMP_WST == 0 werden diejenigen Fälle NICHT zur Berechnung der W.keiten
+ # verwendet, bei denen die WST bereits imputiert worden sind
+ b <- subset(datensatz, ISCO==isco & f02x==geschlecht & !is.na(WST) & STUDIE == 1
+  & IMP_WST==0 )
+ print("Anzahl der Fälle mit gleichem ISCO und gleichem Geschlecht")
+ print(nrow(b))
+
+ if(nrow(b)>0){ # Wenn es noch andere Fälle mit gleichem ISCO und WST gibt
+ print("b groesser als 0 also auf Geschlecht und ISCO bedingen")
+ table1 <- table(b$WST)
+ print("Table der Wochenstunden bedingt auf Geschlecht und ISCO")
+ print(table1)
+ prob1 <- prop.table(as.array(table1))  # Wahrscheinlichkeiten für WST berechnen
+ print("Wahrscheinlichkeiten der Wochenstunden bedingt auf Geschlecht und ISCO")
+ print(prob1)
+ # dann aus WST mit diesen Wahrscheinlichkeiten ziehen
+ }
+
+ if(nrow(b)==0){ # also wenn es keinen entsprechenden Fall mit gleichem ISCO und
+ # gleichem Geschlecht gibt bei dem die WST fehlen (d.h. WST fehlen nur in diesem
+ # einen Fall mit diesem ISCO und diesem Geschlecht)
+ print("b gleich 0 also nur auf Geschlecht bedingen")
+ # aus großem Datensatz alle mit gleichem Geschlecht ziehen, durch
+ # IMP_WST == 0 werden diejenigen Fälle NICHT zur Berechnung der W.keiten
+ # verwendet, bei denen die WST bereits imputiert worden sind
+ c <- subset(datensatz, f02x==geschlecht & !is.na(WST) & STUDIE == 1
+ & IMP_WST==0 )
+ table1 <- table(c$WST)
+ print("Table der Wochenstunden bedingt auf Geschlecht")
+ print(table1)
+ prob1 <- prop.table(as.array(table1))  # Wahrscheinlichkeiten für WST berechnen
+ print("Wahrscheinlichkeiten der Wochenstunden bedingt auf Geschlecht")
+ print(prob1)
+ # dann aus WST mit diesen Wahrscheinlichkeiten ziehen
+ }
+ # Wochenstundenwert für die Imputation ziehen mit der Funktion sample
+ imput <- sample(names(prob1), size = 1, replace=TRUE, prob = prob1)
+ # Im Datensatz imputieren, der die Daten aus SOLAR I und SOLAR II enthält
```

```
+ print("Wochenstunden-Wert der an dieser Stelle imputiert werden soll")
+ print(imput)
+ # Index der Zeile (index) in 1.Spalte des Datensatzes imput_werte1 speichern
+ imput_werte1$index[j] <- index
+ # Wochenstundenwert (imput) in 2.Spalte des Datensatzes imput_werte1 speichern
+ imput_werte1$wochenstunden[j] <- imput
+ print("Matrix,1.Spalte:Index der Zeile,2.Spalte:Zu imputierende Wochenstunden")
+ print(imput_werte1)
+ # Jetzt den Wert im "großen" Datensatz imputieren
+ for(k in 1:nrow(datensatz)){
+ # Wenn der Index übereinstimmt
+ if(datensatz$index[k] == imput_werte1$index[j]){
+ # Zu imputierende Wochenstunden in der Zeile mit diesem Index imputieren
+ datensatz$WST[k] <- as.numeric(imput_werte1$wochenstunden[j])
+ print("Wochenstunden-Wert der imputiert wird")
+ print(datensatz$WST[k])
+ # Indikator IMP_WST auf 1 setzen, d.h. die Wochenstunden werden imputiert
+ datensatz$IMP_WST[k] <- 1
+ }
+ }
+ j <- j+1
+ }
+
+ ############################## SOLAR II #####################################
+ print("***************Imputation der Wochenstunden in SOLAR II***************")
+ a2 <- subset(datensatz_s2, is.na(WST) & kuenstl_geloescht == 1) # hier nur die
+ # Zeilen drin in denen die WST künstlich gelöscht wurden
+
+ # Datensatz anlegen, in den jeweils index der Zeile und imputierte WST
+ # geschrieben werden (1.Spalte: index, 2.Spalte: wochenstunden)
+ imput_werte2 <- data.frame(index=numeric(nrow(a2)),
+ wochenstunden = numeric(nrow(a2)))
+
+ # j auf 1 setzen
+ j <- 1
+
+ # Startwert setzen (der der Funktion als Argument übergeben wurde)
+ set.seed(startwert)
+
+ for (i in 1:nrow(a2)){
+ print("Nächste Stelle i")
+ # Geschlecht,ISCO und Index an der i-ten Stelle aus subset betrachten
+ isco <- a2$ISCO[i]
+ print("ISCO an der Stelle i")
+ print(isco)
+ geschlecht <- a2$f02x[i]
+ print("Geschlecht an der Stelle i")
+ print(geschlecht)
+ index <- a2$index[i]
+ print("Index an der Stelle i")
+ print(index)
+ # aus großen Datensatz alle mit gleichem ISCO und Geschlecht ziehen, durch
+ # IMP_WST == 0 werden diejenigen Fälle NICHT zur Berechnung der W.keiten
+ # verwendet, bei denen die WST bereits imputiert worden sind
+ b <- subset(datensatz, ISCO==isco & f02x==geschlecht & !is.na(WST) & STUDIE == 2
+  & IMP_WST==0 )
+ print("Anzahl der Fälle mit gleichem ISCO und gleichem Geschlecht")
+ print(nrow(b))
+
+ if(nrow(b)>0){ # Wenn es noch andere Fälle mit gleichem ISCO und WST gibt
+ print("b groesser als 0 also auf Geschlecht und ISCO bedingen")
+ table1 <- table(b$WST)
+ print("Table der Wochenstunden bedingt auf Geschlecht und ISCO")
+ print(table1)
+ prob1 <- prop.table(as.array(table1))  # Wahrscheinlichkeiten für WST berechnen
+ print("Wahrscheinlichkeiten der Wochenstunden bedingt auf Geschlecht und ISCO")
+ print(prob1)
+ # dann aus WST mit diesen Wahrscheinlichkeiten ziehen
+ }
+
+ if(nrow(b)==0){ # also wenn es keinen entsprechenden Fall mit gleichem ISCO und
+ # gleichem Geschlecht gibt bei dem die WST fehlen (d.h. WST fehlen nur in diesem
+ # einen Fall mit diesem ISCO und diesem Geschlecht)
```

```
+ print("b gleich 0 also nur auf Geschlecht bedingen")
+ # aus großem Datensatz alle mit gleichem Geschlecht ziehen, durch
+ # IMP_WST == 0 werden diejenigen Fälle NICHT zur Berechnung der W.keiten
+ # verwendet, bei denen die WST bereits imputiert worden sind
+ c <- subset(datensatz, f02x==geschlecht & !is.na(WST) & STUDIE == 2
+ & IMP_WST==0 )
+ table1 <- table(c$WST)
+ print("Table der Wochenstunden bedingt auf Geschlecht")
+ print(table1)
+ prob1 <- prop.table(as.array(table1))  # Wahrscheinlichkeiten für WST berechnen
+ print("Wahrscheinlichkeiten der Wochenstunden bedingt auf Geschlecht")
+ print(prob1)
+ # dann aus WST mit diesen Wahrscheinlichkeiten ziehen
+ }
+ # Wochenstundenwert für die Imputation ziehen mit der Funktion sample
+ imput <- sample(names(prob1), size = 1, replace=TRUE, prob = prob1)
+ # Im Datensatz imputieren, der die Daten aus SOLAR I und SOLAR II enthält
+ print("Wochenstunden-Wert der an dieser Stelle imputiert werden soll")
+ print(imput)
+ # Index der Zeile (index) in 1.Spalte des Datensatzes imput_werte2 speichern
+ imput_werte2$index[j] <- index
+ # Wochenstundenwert (imput) in 2.Spalte des Datensatzes imput_werte2 speichern
+ imput_werte2$wochenstunden[j] <- imput
+ print("Matrix,1.Spalte:Index der Zeile,2.Spalte:Zu imputierende Wochenstunden")
+ print(imput_werte2)
+ # Jetzt den Wert im "großen" Datensatz imputieren
+ for(k in 1:nrow(datensatz)){
+ # Wenn der Index übereinstimmt
+ if(datensatz$index[k] == imput_werte2$index[j]){
+ # Zu imputierende Wochenstunden in der Zeile mit diesem Index imputieren
+ datensatz$WST[k] <- as.numeric(imput_werte2$wochenstunden[j])
+ print("Wochenstunden-Wert der imputiert wird")
+ print(datensatz$WST[k])
+ # Indikator IMP_WST auf 1 setzen, d.h. die Wochenstunden werden imputiert
+ datensatz$IMP_WST[k] <- 1
+ }
+ }
+ j <- j+1
+ }
+
+ ##############################################################################
+ ####                  Anfangsjahr (ANF_JAHR) imputieren                    ####
+ ##############################################################################
+
+ # Indikator, der angibt ob das ANF_JAHR imputiert wurde anlegen
+ datensatz$IMP_AJ <- 0
+
+ ############################### SOLAR I ########################################
+ # in SOLAR I nur bedingen auf SES_r
+ print("**************Imputation des Anfangsjahrs in SOLAR I**************")
+ a3 <- subset(datensatz_s1, is.na(ANF_JAHR) & kuenstl_geloescht == 1) # hier nur
+ # die Zeilen drin in denen das ANF_JAHR künstlich gelöscht wurden
+
+ # Datensatz anlegen, in den jeweils index der Zeile und imputiertes ANF_JAHR
+ # geschrieben werden (1.Spalte: index, 2.Spalte: anfangsjahr)
+
+ imput_werte3 <- data.frame(index=numeric(nrow(a3)),
+ anfangsjahr = numeric(nrow(a3)))
+
+ # j auf 1 setzen
+ j <- 1
+
+ # Startwert setzen (der der Funktion als Argument übergeben wurde)
+ set.seed(startwert)
+
+ for (i in 1: nrow(a3)){
+ print("Nächste Stelle i")
+ # SES und Index an der i-ten Stelle aus subset betrachten
+ ses <- a3$SES_r[i]
+ print("SES an der Stelle i")
+ print(ses)
+ index <- a3$index[i]
+ print("Index an der Stelle i")
```

```
+ print(index)
+ # aus großen Datensatz alle mit gleichem SES ziehen, durch
+ # IMP_AJ == 0 werden diejenigen Fälle NICHT zur Berechnung der W.keiten
+ # verwendet, bei denen das ANF_JAHR bereits imputiert wurde
+ b <- subset(datensatz, SES_r==ses & !is.na(ANF_JAHR) & STUDIE == 1
+  & IMP_AJ==0 )
+ print("Anzahl der Fälle mit gleichem SES")
+ print(nrow(b))
+ # Hier gibt es auf jeden Fall noch andere Fälle mit gleichem SES ! D.h.
+ # nrow(b) > 0 immer !
+ table1 <- table(b$ANF_JAHR)
+ print("Table der Anfangsjahre bedingt auf SES_r")
+ print(table1)
+ prob1 <- prop.table(as.array(table1))  # Wahrscheinlichkeiten für ANF_JAHR
+ # berechnen
+ print("Wahrscheinlichkeiten der Anfangsjahre bedingt auf SES_r")
+ print(prob1)
+ # dann aus den Anfangsjahren mit diesen Wahrscheinlichkeiten ziehen
+ # Anfangsjahr für die Imputation ziehen mit der Funktion sample
+ imput <- sample(names(prob1), size = 1, replace=TRUE, prob = prob1)
+ # Im Datensatz imputieren, der die Daten aus SOLAR I und SOLAR II enthält
+ print("Anfangsjahr das an dieser Stelle imputiert werden soll")
+ print(imput)
+ # Index der Zeile (index) in 1.Spalte des Datensatzes imput_werte3 speichern
+ imput_werte3$index[j] <- index
+ # Anfangsjahr (imput) in 2.Spalte des Datensatzes imput_werte3 speichern
+ imput_werte3$anfangsjahr[j] <- imput
+ print("Matrix,1.Spalte:Index der Zeile,2.Spalte:Zu imputierendes Anfangsjahr")
+ print(imput_werte3)
+ # Jetzt den Wert im "großen" Datensatz imputieren
+ for(k in 1:nrow(datensatz)){
+ # Wenn der Index übereinstimmt
+ if(datensatz$index[k] == imput_werte3$index[j]){
+ # Zu imputierendes Anfangsjahr in der Zeile mit diesem Index imputieren
+ datensatz$ANF_JAHR[k] <- as.numeric(imput_werte3$anfangsjahr[j])
+ print("Anfangsjahr das imputiert wird")
+ print(datensatz$ANF_JAHR[k])
+ # Indikator IMP_AJ auf 1 setzen, d.h. das Anfangsjahr werden imputiert
+ datensatz$IMP_AJ[k] <- 1
+ }
+ }
+ j <- j+1
+ }
+
+ ################################ SOLAR II #####################################
+ # In SOLAR II bedingen auf s2BERUF und SES_r
+ print("**************Imputation des Anfangsjahrs in SOLAR II**************")
+ a4 <- subset(datensatz_s2, is.na(ANF_JAHR) & kuenstl_geloescht == 1) # hier nur
+ # die Zeilen drin in denen das ANF_JAHR künstlich gelöscht wurde
+
+ # Datensatz anlegen, in den jeweils index der Zeile und imputiertes ANF_JAHR
+ # geschrieben werden (1.Spalte: index, 2.Spalte: anfangsjahr)
+ imput_werte4 <- data.frame(index=numeric(nrow(a4)),
+ anfangsjahr = numeric(nrow(a4)))
+
+ # j auf 1 setzen
+ j <- 1
+
+ # Startwert setzen (der der Funktion als Argument übergeben wurde)
+ set.seed(startwert)
+
+ for (i in 1:nrow(a4)){
+ print("Nächste Stelle i")
+ # Geschlecht,ISCO und Index an der i-ten Stelle aus subset betrachten
+ beruf <- a4$s2BERUF[i]
+ print("s2BERUF an der Stelle i")
+ print(beruf)
+ ses <- a4$SES_r[i]
+ print("SES an der Stelle i")
+ print(ses)
+ index <- a4$index[i]
+ print("Index an der Stelle i")
+ print(index)
```

```
+ # aus großen Datensatz alle mit gleichem s2BERUF und SES_r ziehen, durch
+ # IMP_AJ == 0 werden diejenigen Fälle NICHT zur Berechnung der W.keiten
+ # verwendet, bei denen das ANF_JAHR bereits imputiert worden sind
+ b <- subset(datensatz, SES_r==ses & s2BERUF==beruf & !is.na(ANF_JAHR)
+ & STUDIE == 2 & IMP_AJ==0 )
+ print("Anzahl der Fälle mit gleichem s2BERUF und gleichem SES_r")
+ print(nrow(b))
+
+ if(nrow(b)>0){ # Wenn es noch andere Fälle mit gleichem s2BERUF und SES_r gibt
+ print("b groesser als 0 also auf s2BERUF und SES_r bedingen")
+ table1 <- table(b$ANF_JAHR)
+ print("Table der Anfangsjahre bedingt auf s2BERUF und SES_r")
+ print(table1)
+ prob1 <- prop.table(as.array(table1))  # Wahrscheinlichkeiten für ANF_JAHR
+ # berechnen
+ print("Wahrscheinlichkeiten der Anfangsjahre bedingt auf s2BERUF und SES_r")
+ print(prob1)
+ # dann aus ANF_JAHR mit diesen Wahrscheinlichkeiten ziehen
+ }
+
+ if(nrow(b)==0){ # also wenn es keinen entsprechenden Fall mit gleichem s2BERUF,
+ # gleichem SES_r gibt bei dem das ANF_JAHR fehlt (d.h. ANF_JAHR fehlt nur in
+ # diesem einen Fall mit diesem s2BERUF und diesem SES_r)
+ print("b gleich 0 also nur auf SES_r bedingen")
+ # aus großem Datensatz alle mit gleichem SES_r ziehen, durch
+ # IMP_AJ == 0 werden diejenigen Fälle NICHT zur Berechnung der W.keiten
+ # verwendet, bei denen das ANF_JAHR bereits imputiert wurde
+ c <- subset(datensatz, SES_r==ses & !is.na(ANF_JAHR) & STUDIE == 2
+ & IMP_AJ==0 )
+ table1 <- table(c$ANF_JAHR)
+ print("Table der Anfangsjahre bedingt auf Geschlecht")
+ print(table1)
+ prob1 <- prop.table(as.array(table1))  # Wahrscheinlichkeiten für ANF_JAHR
+ # berechnen
+ print("Wahrscheinlichkeiten der Anfangsjahre bedingt auf Geschlecht")
+ print(prob1)
+ # dann aus ANF_JAHR mit diesen Wahrscheinlichkeiten ziehen
+ }
+ # Anfangsjahr für die Imputation ziehen mit der Funktion sample
+ imput <- sample(names(prob1), size = 1, replace=TRUE, prob = prob1)
+ # Im Datensatz imputieren, der die Daten aus SOLAR I und SOLAR II enthält
+ print("Anfangsjahr das an dieser Stelle imputiert werden soll")
+ print(imput)
+ # Index der Zeile (index) in 1.Spalte des Datensatzes imput_werte4 speichern
+ imput_werte4$index[j] <- index
+ # Anfangsjahr (imput) in 2.Spalte des Datensatzes imput_werte4 speichern
+ imput_werte4$anfangsjahr[j] <- imput
+ print("Matrix,1.Spalte:Index der Zeile,2.Spalte:Zu imputierendes Anfangsjahr")
+ print(imput_werte4)
+ # Jetzt den Wert im "großen" Datensatz imputieren
+ for(k in 1:nrow(datensatz)){
+ # Wenn der Index übereinstimmt
+ if(datensatz$index[k] == imput_werte4$index[j]){
+   # Zu imputierendes Anfangsjahr in der Zeile mit diesem Index imputieren
+   datensatz$ANF_JAHR[k] <- as.numeric(imput_werte4$anfangsjahr[j])
+   print("Anfangsjahr das imputiert wird")
+   print(datensatz$ANF_JAHR[k])
+   # Indikator IMP_AJ auf 1 setzen, d.h. das Anfangsjahr wurde imputiert
+   datensatz$IMP_AJ[k] <- 1
+   }
+   }
+   j <- j+1
+   # immer: j um 1 erhöhen, d.h. der nächste Wert im Vektor wird betrachet
+   }
+
+ ##############################################################################
+ ####                  Endjahr (END_JAHRx) imputieren                      ####
+ ##############################################################################
+
+ # Indikator, der angibt ob das END_JAHRx imputiert wurde anlegen
+ datensatz$IMP_EJ <- 0
+
+ ############################## SOLAR I ########################################
```

```
+ # in SOLAR I nur bedingen auf SES_r
+ print("***************Imputation des Endjahrs in SOLAR I***************")
+ a7 <- subset(datensatz_s1, is.na(END_JAHRx) & kuenstl_geloescht == 1) # hier nur
+ # die Zeilen drin in denen das END_JAHRx künstlich gelöscht wurden
+
+ # Datensatz anlegen, in den jeweils index der Zeile und imputiertes END_JAHRx
+ # geschrieben werden (1.Spalte: index, 2.Spalte: endjahr)
+ imput_werte7 <- data.frame(index=numeric(nrow(a7)),
+ endjahr1 = numeric(nrow(a7)), endjahr2 = numeric(nrow(a7)),
+ endjahr3 = numeric(nrow(a7)), endjahr4 = numeric(nrow(a7)),
+ endjahr5 = numeric(nrow(a7)), endjahr6 = numeric(nrow(a7)),
+ endjahr7 = numeric(nrow(a7)), endjahr8 = numeric(nrow(a7)),
+ endjahr9 = numeric(nrow(a7)), endjahr10 = numeric(nrow(a7)),
+ endjahr11 = numeric(nrow(a7)), endjahr12 = numeric(nrow(a7)),
+ endjahr13 = numeric(nrow(a7)), endjahr14 = numeric(nrow(a7)),
+ endjahr15 = numeric(nrow(a7)), endjahr16 = numeric(nrow(a7)),
+ endjahr17 = numeric(nrow(a7)), endjahr18 = numeric(nrow(a7)),
+ endjahr19 = numeric(nrow(a7)), endjahr20 = numeric(nrow(a7))
+ )
+
+ # j auf 1 setzen
+ j <- 1
+
+ # Startwert setzen (der der Funktion als Argument übergeben wurde)
+ set.seed(startwert)
+
+ for (i in 1: nrow(a7)){
+ print("Nächste Stelle i")
+ # SES und Index an der i-ten Stelle aus subset betrachten
+ ses <- a7$SES_r[i]
+ print("SES an der Stelle i")
+ print(ses)
+ index <- a7$index[i]
+ print("Index an der Stelle i")
+ print(index)
+ # aus großen Datensatz alle mit gleichem SES ziehen, durch
+ # IMP_EJ == 0 werden diejenigen Fälle NICHT zur Berechnung der W.keiten
+ # verwendet, bei denen das END_JAHRx bereits imputiert wurde
+ b <- subset(datensatz, SES_r==ses & !is.na(END_JAHRx) & STUDIE == 1
+  & IMP_EJ==0 )
+ print("Anzahl der Fälle mit gleichem SES")
+ print(nrow(b))
+ # Hier gibt es auf jeden Fall noch andere Fälle mit gleichem SES ! D.h.
+ # nrow(b) > 0 immer !
+ table1 <- table(b$END_JAHRx)
+ print("Table der Endjahre bedingt auf SES_r")
+ print(table1)
+ prob1 <- prop.table(as.array(table1))  # Wahrscheinlichkeiten für END_JAHRx
+ # berechnen
+ print("Wahrscheinlichkeiten der Endjahre bedingt auf SES_r")
+ print(prob1)
+ # dann aus den Endjahren mit diesen Wahrscheinlichkeiten ziehen
+ # Endjahr für die Imputation ziehen mit der Funktion sample
+ imput <- as.list(sample(names(prob1), size = 20, replace=TRUE, prob = prob1))
+ # Im Datensatz imputieren, der die Daten aus SOLAR I und SOLAR II enthält
+ print("Endjahr das an dieser Stelle imputiert werden soll")
+ print(imput)
+ # Index der Zeile (index) in 1.Spalte des Datensatzes imput_werte7 speichern
+ imput_werte7$index[j] <- index
+ # Endjahr (imput) in 2.Spalte des Datensatzes imput_werte7 speichern
+ imput_werte7$endjahr1[j] <- as.numeric(imput[[1]])
+ imput_werte7$endjahr2[j] <- as.numeric(imput[[2]])
+ imput_werte7$endjahr3[j] <- as.numeric(imput[[3]])
+ imput_werte7$endjahr4[j] <- as.numeric(imput[[4]])
+ imput_werte7$endjahr5[j] <- as.numeric(imput[[5]])
+ imput_werte7$endjahr6[j] <- as.numeric(imput[[6]])
+ imput_werte7$endjahr7[j] <- as.numeric(imput[[7]])
+ imput_werte7$endjahr8[j] <- as.numeric(imput[[8]])
+ imput_werte7$endjahr9[j] <- as.numeric(imput[[9]])
+ imput_werte7$endjahr10[j] <- as.numeric(imput[[10]])
+ imput_werte7$endjahr11[j] <- as.numeric(imput[[11]])
+ imput_werte7$endjahr12[j] <- as.numeric(imput[[12]])
+ imput_werte7$endjahr13[j] <- as.numeric(imput[[13]])
```

```
+ imput_werte7$endjahr14[j] <- as.numeric(imput[[14]])
+ imput_werte7$endjahr15[j] <- as.numeric(imput[[15]])
+ imput_werte7$endjahr16[j] <- as.numeric(imput[[16]])
+ imput_werte7$endjahr17[j] <- as.numeric(imput[[17]])
+ imput_werte7$endjahr18[j] <- as.numeric(imput[[18]])
+ imput_werte7$endjahr19[j] <- as.numeric(imput[[19]])
+ imput_werte7$endjahr20[j] <- as.numeric(imput[[20]])
+ print("Matrix,1.Spalte:Index der Zeile,2.Spalte:Zu imputierendes Endjahr")
+ print(imput_werte7)
+ # Jetzt den Wert im "großen" Datensatz imputieren
+ for(k in 1:nrow(datensatz)){
+ # Wenn der Index übereinstimmt
+ if(datensatz$index[k] == imput_werte7$index[j]){
+ # Zu imputierendes Endjahr in der Zeile mit diesem Index imputieren
+ if (!is.na(datensatz$ANF_MONAT[k])&!is.na(datensatz$END_MONATx[k])&
+ (datensatz$ANF_MONAT[k] > datensatz$END_MONATx[k])){
+ if (datensatz$ANF_JAHR[k] < imput_werte7$endjahr1[j]){
+ datensatz$END_JAHRx[k] <- as.numeric(imput_werte7$endjahr1[j])
+ print("Endjahr das imputiert wird")
+ print(datensatz$END_JAHRx[k])
+ # Indikator IMP_EJ auf 1 setzen, d.h. das Endjahr wurde imputiert
+ datensatz$IMP_EJ[k] <- 1
+ }
+ else{
+ if (datensatz$ANF_JAHR[k] < imput_werte7$endjahr2[j]){
+ datensatz$END_JAHRx[k] <- as.numeric(imput_werte7$endjahr2[j])
+ print("Endjahr das imputiert wird")
+ print(datensatz$END_JAHRx[k])
+ # Indikator IMP_EJ auf 1 setzen, d.h. das Endjahr wurde imputiert
+ datensatz$IMP_EJ[k] <- 1
+ }
+ else{
+ if (datensatz$ANF_JAHR[k] < imput_werte7$endjahr3[j]){
+ datensatz$END_JAHRx[k] <- as.numeric(imput_werte7$endjahr3[j])
+ print("Endjahr das imputiert wird")
+ print(datensatz$END_JAHRx[k])
+ # Indikator IMP_EJ auf 1 setzen, d.h. das Endjahr wurde imputiert
+ datensatz$IMP_EJ[k] <- 1
+ }
+ else{
+ if (datensatz$ANF_JAHR[k] < imput_werte7$endjahr4[j]){
+ datensatz$END_JAHRx[k] <- as.numeric(imput_werte7$endjahr4[j])
+ print("Endjahr das imputiert wird")
+ print(datensatz$END_JAHRx[k])
+ # Indikator IMP_EJ auf 1 setzen, d.h. das Endjahr wurde imputiert
+ datensatz$IMP_EJ[k] <- 1
+ }
+ else{
+ if (datensatz$ANF_JAHR[k] < imput_werte7$endjahr5[j]){
+ datensatz$END_JAHRx[k] <- as.numeric(imput_werte7$endjahr5[j])
+ print("Endjahr das imputiert wird")
+ print(datensatz$END_JAHRx[k])
+ # Indikator IMP_EJ auf 1 setzen, d.h. das Endjahr wurde imputiert
+ datensatz$IMP_EJ[k] <- 1
+ }
+ else{
+ if (datensatz$ANF_JAHR[k] < imput_werte7$endjahr6[j]){
+ datensatz$END_JAHRx[k] <- as.numeric(imput_werte7$endjahr6[j])
+ print("Endjahr das imputiert wird")
+ print(datensatz$END_JAHRx[k])
+ # Indikator IMP_EJ auf 1 setzen, d.h. das Endjahr wurde imputiert
+ datensatz$IMP_EJ[k] <- 1
+ }
+ else{
+ if (datensatz$ANF_JAHR[k] < imput_werte7$endjahr7[j]){
+ datensatz$END_JAHRx[k] <- as.numeric(imput_werte7$endjahr7[j])
+ print("Endjahr das imputiert wird")
+ print(datensatz$END_JAHRx[k])
+ # Indikator IMP_EJ auf 1 setzen, d.h. das Endjahr wurde imputiert
+ datensatz$IMP_EJ[k] <- 1
+ }
+ else{
+ if (datensatz$ANF_JAHR[k] < imput_werte7$endjahr8[j]){
```

```
+ datensatz$END_JAHRx[k] <- as.numeric(imput_werte7$endjahr8[j])
+ print("Endjahr das imputiert wird")
+ print(datensatz$END_JAHRx[k])
+ # Indikator IMP_EJ auf 1 setzen, d.h. das Endjahr wurde imputiert
+ datensatz$IMP_EJ[k] <- 1
+ }
+ else{
+ if (datensatz$ANF_JAHR[k] < imput_werte7$endjahr9[j]){
+ datensatz$END_JAHRx[k] <- as.numeric(imput_werte7$endjahr9[j])
+ print("Endjahr das imputiert wird")
+ print(datensatz$END_JAHRx[k])
+ # Indikator IMP_EJ auf 1 setzen, d.h. das Endjahr wurde imputiert
+ datensatz$IMP_EJ[k] <- 1
+ }
+ else{
+ if (datensatz$ANF_JAHR[k] < imput_werte7$endjahr10[j]){
+ datensatz$END_JAHRx[k] <- as.numeric(imput_werte7$endjahr10[j])
+ print("Endjahr das imputiert wird")
+ print(datensatz$END_JAHRx[k])
+ # Indikator IMP_EJ auf 1 setzen, d.h. das Endjahr wurde imputiert
+ datensatz$IMP_EJ[k] <- 1
+ }
+ else{
+ if (datensatz$ANF_JAHR[k] < imput_werte7$endjahr11[j]){
+ datensatz$END_JAHRx[k] <- as.numeric(imput_werte7$endjahr11[j])
+ print("Endjahr das imputiert wird")
+ print(datensatz$END_JAHRx[k])
+ # Indikator IMP_EJ auf 1 setzen, d.h. das Endjahr wurde imputiert
+ datensatz$IMP_EJ[k] <- 1
+ }
+ else{
+ if (datensatz$ANF_JAHR[k] < imput_werte7$endjahr12[j]){
+ datensatz$END_JAHRx[k] <- as.numeric(imput_werte7$endjahr12[j])
+ print("Endjahr das imputiert wird")
+ print(datensatz$END_JAHRx[k])
+ # Indikator IMP_EJ auf 1 setzen, d.h. das Endjahr wurde imputiert
+ datensatz$IMP_EJ[k] <- 1
+ }
+ else{
+ if (datensatz$ANF_JAHR[k] < imput_werte7$endjahr13[j]){
+ datensatz$END_JAHRx[k] <- as.numeric(imput_werte7$endjahr13[j])
+ print("Endjahr das imputiert wird")
+ print(datensatz$END_JAHRx[k])
+ # Indikator IMP_EJ auf 1 setzen, d.h. das Endjahr wurde imputiert
+ datensatz$IMP_EJ[k] <- 1
+ }
+ else{
+ if (datensatz$ANF_JAHR[k] < imput_werte7$endjahr14[j]){
+ datensatz$END_JAHRx[k] <- as.numeric(imput_werte7$endjahr14[j])
+ print("Endjahr das imputiert wird")
+ print(datensatz$END_JAHRx[k])
+ # Indikator IMP_EJ auf 1 setzen, d.h. das Endjahr wurde imputiert
+ datensatz$IMP_EJ[k] <- 1
+ }
+ else{
+ if (datensatz$ANF_JAHR[k] < imput_werte7$endjahr15[j]){
+ datensatz$END_JAHRx[k] <- as.numeric(imput_werte7$endjahr15[j])
+ print("Endjahr das imputiert wird")
+ print(datensatz$END_JAHRx[k])
+ # Indikator IMP_EJ auf 1 setzen, d.h. das Endjahr wurde imputiert
+ datensatz$IMP_EJ[k] <- 1
+ }
+ else{
+ if (datensatz$ANF_JAHR[k] < imput_werte7$endjahr16[j]){
+ datensatz$END_JAHRx[k] <- as.numeric(imput_werte7$endjahr16[j])
+ print("Endjahr das imputiert wird")
+ print(datensatz$END_JAHRx[k])
+ # Indikator IMP_EJ auf 1 setzen, d.h. das Endjahr wurde imputiert
+ datensatz$IMP_EJ[k] <- 1
+ }
+ else{
+ if (datensatz$ANF_JAHR[k] < imput_werte7$endjahr17[j]){
+ datensatz$END_JAHRx[k] <- as.numeric(imput_werte7$endjahr17[j])
```

```
+ print("Endjahr das imputiert wird")
+ print(datensatz$END_JAHRx[k])
+ # Indikator IMP_EJ auf 1 setzen, d.h. das Endjahr wurde imputiert
+ datensatz$IMP_EJ[k] <- 1
+ }
+ else{
+ if (datensatz$ANF_JAHR[k] < imput_werte7$endjahr18[j]){
+ datensatz$END_JAHRx[k] <- as.numeric(imput_werte7$endjahr18[j])
+ print("Endjahr das imputiert wird")
+ print(datensatz$END_JAHRx[k])
+ # Indikator IMP_EJ auf 1 setzen, d.h. das Endjahr wurde imputiert
+ datensatz$IMP_EJ[k] <- 1
+ }
+ else{
+ if (datensatz$ANF_JAHR[k] < imput_werte7$endjahr19[j]){
+ datensatz$END_JAHRx[k] <- as.numeric(imput_werte7$endjahr19[j])
+ print("Endjahr das imputiert wird")
+ print(datensatz$END_JAHRx[k])
+ # Indikator IMP_EJ auf 1 setzen, d.h. das Endjahr wurde imputiert
+ datensatz$IMP_EJ[k] <- 1
+ }
+ else{
+ if (datensatz$ANF_JAHR[k] < imput_werte7$endjahr20[j]){
+ datensatz$END_JAHRx[k] <- as.numeric(imput_werte7$endjahr20[j])
+ print("Endjahr das imputiert wird")
+ print(datensatz$END_JAHRx[k])
+ # Indikator IMP_EJ auf 1 setzen, d.h. das Endjahr wurde imputiert
+ datensatz$IMP_EJ[k] <- 1
+ }
+ }}}}}}}}}}
+ }
+ }
+ }
+ }
+ }
+ }
+ }
+ }
+ }
+ if (is.na(datensatz$ANF_MONAT[k]) | is.na(datensatz$END_MONATx[k]) |
+ !is.na(datensatz$ANF_MONAT[k])&!is.na(datensatz$END_MONATx[k])
+ & (datensatz$ANF_MONAT[k] <= datensatz$END_MONATx[k])){
+ if (datensatz$ANF_JAHR[k] <= imput_werte7$endjahr1[j]){
+ datensatz$END_JAHRx[k] <- as.numeric(imput_werte7$endjahr1[j])
+ print("Endjahr das imputiert wird")
+ print(datensatz$END_JAHRx[k])
+ # Indikator IMP_EJ auf 1 setzen, d.h. das Endjahr wurde imputiert
+ datensatz$IMP_EJ[k] <- 1
+ }
+ else{
+ if (datensatz$ANF_JAHR[k] <= imput_werte7$endjahr2[j]){
+ datensatz$END_JAHRx[k] <- as.numeric(imput_werte7$endjahr2[j])
+ print("Endjahr das imputiert wird")
+ print(datensatz$END_JAHRx[k])
+ # Indikator IMP_EJ auf 1 setzen, d.h. das Endjahr wurde imputiert
+ datensatz$IMP_EJ[k] <- 1
+ }
+ else{
+ if (datensatz$ANF_JAHR[k] <= imput_werte7$endjahr3[j]){
+ datensatz$END_JAHRx[k] <- as.numeric(imput_werte7$endjahr3[j])
+ print("Endjahr das imputiert wird")
+ print(datensatz$END_JAHRx[k])
+ # Indikator IMP_EJ auf 1 setzen, d.h. das Endjahr wurde imputiert
+ datensatz$IMP_EJ[k] <- 1
+ }
+ else{
+ if (datensatz$ANF_JAHR[k] <= imput_werte7$endjahr4[j]){
+ datensatz$END_JAHRx[k] <- as.numeric(imput_werte7$endjahr4[j])
+ print("Endjahr das imputiert wird")
+ print(datensatz$END_JAHRx[k])
+ # Indikator IMP_EJ auf 1 setzen, d.h. das Endjahr wurde imputiert
+ datensatz$IMP_EJ[k] <- 1
```

```
+ }
+ else{
+ if (datensatz$ANF_JAHR[k] <= imput_werte7$endjahr5[j]){
+ datensatz$END_JAHRx[k] <- as.numeric(imput_werte7$endjahr5[j])
+ print("Endjahr das imputiert wird")
+ print(datensatz$END_JAHRx[k])
+ # Indikator IMP_EJ auf 1 setzen, d.h. das Endjahr wurde imputiert
+ datensatz$IMP_EJ[k] <- 1
+ }
+ else{
+ if (datensatz$ANF_JAHR[k] <= imput_werte7$endjahr6[j]){
+ datensatz$END_JAHRx[k] <- as.numeric(imput_werte7$endjahr6[j])
+ print("Endjahr das imputiert wird")
+ print(datensatz$END_JAHRx[k])
+ # Indikator IMP_EJ auf 1 setzen, d.h. das Endjahr wurde imputiert
+ datensatz$IMP_EJ[k] <- 1
+ }
+ else{
+ if (datensatz$ANF_JAHR[k] <= imput_werte7$endjahr7[j]){
+ datensatz$END_JAHRx[k] <- as.numeric(imput_werte7$endjahr7[j])
+ print("Endjahr das imputiert wird")
+ print(datensatz$END_JAHRx[k])
+ # Indikator IMP_EJ auf 1 setzen, d.h. das Endjahr wurde imputiert
+ datensatz$IMP_EJ[k] <- 1
+ }
+ else{
+ if (datensatz$ANF_JAHR[k] <= imput_werte7$endjahr8[j]){
+ datensatz$END_JAHRx[k] <- as.numeric(imput_werte7$endjahr8[j])
+ print("Endjahr das imputiert wird")
+ print(datensatz$END_JAHRx[k])
+ # Indikator IMP_EJ auf 1 setzen, d.h. das Endjahr wurde imputiert
+ datensatz$IMP_EJ[k] <- 1
+ }
+ else{
+ if (datensatz$ANF_JAHR[k] <= imput_werte7$endjahr9[j]){
+ datensatz$END_JAHRx[k] <- as.numeric(imput_werte7$endjahr9[j])
+ print("Endjahr das imputiert wird")
+ print(datensatz$END_JAHRx[k])
+ # Indikator IMP_EJ auf 1 setzen, d.h. das Endjahr wurde imputiert
+ datensatz$IMP_EJ[k] <- 1
+ }
+ else{
+ if (datensatz$ANF_JAHR[k] <= imput_werte7$endjahr10[j]){
+ datensatz$END_JAHRx[k] <- as.numeric(imput_werte7$endjahr10[j])
+ print("Endjahr das imputiert wird")
+ print(datensatz$END_JAHRx[k])
+ # Indikator IMP_EJ auf 1 setzen, d.h. das Endjahr wurde imputiert
+ datensatz$IMP_EJ[k] <- 1
+ }
+ else{
+ if (datensatz$ANF_JAHR[k] <= imput_werte7$endjahr11[j]){
+ datensatz$END_JAHRx[k] <- as.numeric(imput_werte7$endjahr11[j])
+ print("Endjahr das imputiert wird")
+ print(datensatz$END_JAHRx[k])
+ # Indikator IMP_EJ auf 1 setzen, d.h. das Endjahr wurde imputiert
+ datensatz$IMP_EJ[k] <- 1
+ }
+ else{
+ if (datensatz$ANF_JAHR[k] <= imput_werte7$endjahr12[j]){
+ datensatz$END_JAHRx[k] <- as.numeric(imput_werte7$endjahr12[j])
+ print("Endjahr das imputiert wird")
+ print(datensatz$END_JAHRx[k])
+ # Indikator IMP_EJ auf 1 setzen, d.h. das Endjahr wurde imputiert
+ datensatz$IMP_EJ[k] <- 1
+ }
+ else{
+ if (datensatz$ANF_JAHR[k] <= imput_werte7$endjahr13[j]){
+ datensatz$END_JAHRx[k] <- as.numeric(imput_werte7$endjahr13[j])
+ print("Endjahr das imputiert wird")
+ print(datensatz$END_JAHRx[k])
+ # Indikator IMP_EJ auf 1 setzen, d.h. das Endjahr wurde imputiert
+ datensatz$IMP_EJ[k] <- 1
+ }
```

```
+ else{
+ if (datensatz$ANF_JAHR[k] <= imput_werte7$endjahr14[j]){
+ datensatz$END_JAHRx[k] <- as.numeric(imput_werte7$endjahr14[j])
+ print("Endjahr das imputiert wird")
+ print(datensatz$END_JAHRx[k])
+ # Indikator IMP_EJ auf 1 setzen, d.h. das Endjahr wurde imputiert
+ datensatz$IMP_EJ[k] <- 1
+ }
+ else{
+ if (datensatz$ANF_JAHR[k] <= imput_werte7$endjahr15[j]){
+ datensatz$END_JAHRx[k] <- as.numeric(imput_werte7$endjahr15[j])
+ print("Endjahr das imputiert wird")
+ print(datensatz$END_JAHRx[k])
+ # Indikator IMP_EJ auf 1 setzen, d.h. das Endjahr wurde imputiert
+ datensatz$IMP_EJ[k] <- 1
+ }
+ else{
+ if (datensatz$ANF_JAHR[k] <= imput_werte7$endjahr16[j]){
+ datensatz$END_JAHRx[k] <- as.numeric(imput_werte7$endjahr16[j])
+ print("Endjahr das imputiert wird")
+ print(datensatz$END_JAHRx[k])
+ # Indikator IMP_EJ auf 1 setzen, d.h. das Endjahr wurde imputiert
+ datensatz$IMP_EJ[k] <- 1
+ }
+ else{
+ if (datensatz$ANF_JAHR[k] <= imput_werte7$endjahr17[j]){
+ datensatz$END_JAHRx[k] <- as.numeric(imput_werte7$endjahr17[j])
+ print("Endjahr das imputiert wird")
+ print(datensatz$END_JAHRx[k])
+ # Indikator IMP_EJ auf 1 setzen, d.h. das Endjahr wurde imputiert
+ datensatz$IMP_EJ[k] <- 1
+ }
+ else{
+ if (datensatz$ANF_JAHR[k] <= imput_werte7$endjahr18[j]){
+ datensatz$END_JAHRx[k] <- as.numeric(imput_werte7$endjahr18[j])
+ print("Endjahr das imputiert wird")
+ print(datensatz$END_JAHRx[k])
+ # Indikator IMP_EJ auf 1 setzen, d.h. das Endjahr wurde imputiert
+ datensatz$IMP_EJ[k] <- 1
+ }
+ else{
+ if (datensatz$ANF_JAHR[k] <= imput_werte7$endjahr19[j]){
+ datensatz$END_JAHRx[k] <- as.numeric(imput_werte7$endjahr19[j])
+ print("Endjahr das imputiert wird")
+ print(datensatz$END_JAHRx[k])
+ # Indikator IMP_EJ auf 1 setzen, d.h. das Endjahr wurde imputiert
+ datensatz$IMP_EJ[k] <- 1
+ }
+ else{
+ if (datensatz$ANF_JAHR[k] <= imput_werte7$endjahr20[j]){
+ datensatz$END_JAHRx[k] <- as.numeric(imput_werte7$endjahr20[j])
+ print("Endjahr das imputiert wird")
+ print(datensatz$END_JAHRx[k])
+ # Indikator IMP_EJ auf 1 setzen, d.h. das Endjahr wurde imputiert
+ datensatz$IMP_EJ[k] <- 1
+ }
+ }}}}}}}}}}
+ }
+ }
+ }
+ }
+ }
+ }
+ }
+ }
+ }
+ }
+ }
+ j <- j+1
+ }
+
+ ############################## SOLAR II #####################################
```

```
+ # in SOLAR II bedingen auf s2BERUF und SES_r
+ print("***************Imputation des Endjahrs in SOLAR II***************")
+ a8 <- subset(datensatz, STUDIE == 2 & is.na(END_JAHRx) & kuenstl_geloescht == 1)
+ # hier nur die Zeilen drin in denen das END_JAHRx künstlich gelöscht wurde
+
+ # Datensatz anlegen, in den jeweils index der Zeile und imputiertes END_JAHRx
+ # geschrieben werden (1.Spalte: index, 2.Spalte: endjahr)
+ imput_werte8 <- data.frame(index=numeric(nrow(a8)),
+ endjahr1 = numeric(nrow(a8)), endjahr2 = numeric(nrow(a8)),
+ endjahr3 = numeric(nrow(a8)), endjahr4 = numeric(nrow(a8)),
+ endjahr5 = numeric(nrow(a8)), endjahr6 = numeric(nrow(a8)),
+ endjahr7 = numeric(nrow(a8)), endjahr8 = numeric(nrow(a8)),
+ endjahr9 = numeric(nrow(a8)), endjahr10 = numeric(nrow(a8)),
+ endjahr11 = numeric(nrow(a8)), endjahr12 = numeric(nrow(a8)),
+ endjahr13 = numeric(nrow(a8)), endjahr14 = numeric(nrow(a8)),
+ endjahr15 = numeric(nrow(a8)), endjahr16 = numeric(nrow(a8)),
+ endjahr17 = numeric(nrow(a8)), endjahr18 = numeric(nrow(a8)),
+ endjahr19 = numeric(nrow(a8)), endjahr20 = numeric(nrow(a8))
+ )
+
+ # j auf 1 setzen
+ j <- 1
+
+ # Startwert setzen (der der Funktion als Argument übergeben wurde)
+ set.seed(startwert)
+
+ for (i in 1: nrow(a8)){
+ print("Nächste Stelle i")
+ # SES und Index an der i-ten Stelle aus subset betrachten
+ beruf <- a8$s2BERUF[i]
+ print("s2BERUF an der Stelle i")
+ print(beruf)
+ ses <- a8$SES_r[i]
+ print("SES an der Stelle i")
+ print(ses)
+ index <- a8$index[i]
+ print("Index an der Stelle i")
+ print(index)
+ # aus großen Datensatz alle mit gleichem s2BERUF SES_r ziehen, durch
+ # IMP_EJ == 0 werden diejenigen Fälle NICHT zur Berechnung der W.keiten
+ # verwendet, bei denen das END_JAHRx bereits imputiert wurde
+ b <- subset(datensatz, s2BERUF == beruf & SES_r==ses & !is.na(END_JAHRx)
+ & STUDIE == 2 & IMP_EJ==0 )
+ print("Anzahl der Fälle mit gleichem SES und s2BERUF")
+ print(nrow(b))
+
+ if(nrow(b)>0){ # Wenn es noch andere Fälle mit gleichem s2BERUF und SES_r gibt
+ print("b groesser als 0 also auf s2BERUF und SES_r bedingen")
+ table1 <- table(b$END_JAHRx)
+ print("Table der Endjahre bedingt auf SES_r und s2BERUF")
+ print(table1)
+ prob1 <- prop.table(as.array(table1))  # Wahrscheinlichkeiten für END_JAHRx
+ # berechnen
+ print("Wahrscheinlichkeiten der Endjahre bedingt auf SES_r und s2BERUF")
+ print(prob1)
+ }
+ if(nrow(b)==0 | (nrow(b)==1 & b$END_JAHRx[1] < a8$ANF_JAHR[i]) |
+ (nrow(b)==2 & b$END_JAHRx[1] < a8$ANF_JAHR[i] & b$END_JAHRx[2] < a8$ANF_JAHR[i])
+ |(nrow(b)==3 & b$END_JAHRx[1] < a8$ANF_JAHR[i] & b$END_JAHRx[2] < a8$ANF_JAHR[i]
+ & b$END_JAHRx[3] < a8$ANF_JAHR[i])){
+ # Wenn es sonst keinen Fall mit gleichem s2BERUF und
+ # gleichem SES_r gibt bei dem das END_JAHRx fehlt (d.h. END_JAHRx fehlt nur in
+ # diesem einen Fall mit diesem s2BERUF und diesem SES_r)
+ # oder es gibt nur 1(2/3) fälle und bei denen wäre dann das Anfangsjahr >
+ # Endjahr
+ print("b gleich 0 also nur auf SES_r bedingen")
+ # aus großem Datensatz alle mit gleichem SES_r ziehen, durch
+ # IMP_EJ == 0 werden diejenigen Fälle NICHT zur Berechnung der W.keiten
+ # verwendet, bei denen das END_JAHRx bereits imputiert wurde
+ c <- subset(datensatz, SES_r==ses & !is.na(END_JAHRx) & STUDIE == 2
+ & IMP_EJ==0 )
+ table1 <- table(c$END_JAHRx)
+ print("Table der Endjahre bedingt auf SES_r")
```

```
+ print(table1)
+ prob1 <- prop.table(as.array(table1))  # Wahrscheinlichkeiten für END_JAHRx
+ # berechnen
+ print("Wahrscheinlichkeiten der Endjahre bedingt auf Geschlecht")
+ print(prob1)
+ }
+ # dann aus den Endjahren mit diesen Wahrscheinlichkeiten ziehen
+ # Endjahr für die Imputation ziehen mit der Funktion sample
+ imput <- as.list(sample(names(prob1), size = 20, replace=TRUE, prob = prob1))
+ # Im Datensatz imputieren, der die Daten aus SOLAR I und SOLAR II enthält
+ print("Endjahr das an dieser Stelle imputiert werden soll")
+ print(imput)
+ # Index der Zeile (index) in 1.Spalte des Datensatzes imput_werte8 speichern
+ imput_werte8$index[j] <- index
+ # Endjahr (imput) in 2.Spalte des Datensatzes imput_werte8 speichern
+ imput_werte8$endjahr1[j] <- as.numeric(imput[[1]])
+ imput_werte8$endjahr2[j] <- as.numeric(imput[[2]])
+ imput_werte8$endjahr3[j] <- as.numeric(imput[[3]])
+ imput_werte8$endjahr4[j] <- as.numeric(imput[[4]])
+ imput_werte8$endjahr5[j] <- as.numeric(imput[[5]])
+ imput_werte8$endjahr6[j] <- as.numeric(imput[[6]])
+ imput_werte8$endjahr7[j] <- as.numeric(imput[[7]])
+ imput_werte8$endjahr8[j] <- as.numeric(imput[[8]])
+ imput_werte8$endjahr9[j] <- as.numeric(imput[[9]])
+ imput_werte8$endjahr10[j] <- as.numeric(imput[[10]])
+ imput_werte8$endjahr11[j] <- as.numeric(imput[[11]])
+ imput_werte8$endjahr12[j] <- as.numeric(imput[[12]])
+ imput_werte8$endjahr13[j] <- as.numeric(imput[[13]])
+ imput_werte8$endjahr14[j] <- as.numeric(imput[[14]])
+ imput_werte8$endjahr15[j] <- as.numeric(imput[[15]])
+ imput_werte8$endjahr16[j] <- as.numeric(imput[[16]])
+ imput_werte8$endjahr17[j] <- as.numeric(imput[[17]])
+ imput_werte8$endjahr18[j] <- as.numeric(imput[[18]])
+ imput_werte8$endjahr19[j] <- as.numeric(imput[[19]])
+ imput_werte8$endjahr20[j] <- as.numeric(imput[[20]])
+ print("Matrix,1.Spalte:Index der Zeile,2.Spalte:Zu imputierendes Endjahr")
+ print(imput_werte8)
+ # Jetzt den Wert im "großen" Datensatz imputieren
+ for(k in 1:nrow(datensatz)){
+ # Wenn der Index übereinstimmt
+ if(datensatz$index[k] == imput_werte8$index[j]){
+ # Zu imputierendes Endjahr in der Zeile mit diesem Index imputieren
+ if (!is.na(datensatz$ANF_MONAT[k])&!is.na(datensatz$END_MONATx[k])&
+ (datensatz$ANF_MONAT[k] > datensatz$END_MONATx[k])){
+ if (datensatz$ANF_JAHR[k] < imput_werte8$endjahr1[j]){
+ datensatz$END_JAHRx[k] <- as.numeric(imput_werte8$endjahr1[j])
+ print("Endjahr das imputiert wird")
+ print(datensatz$END_JAHRx[k])
+ # Indikator IMP_EJ auf 1 setzen, d.h. das Endjahr wurde imputiert
+ datensatz$IMP_EJ[k] <- 1
+ }
+ else{
+ if (datensatz$ANF_JAHR[k] < imput_werte8$endjahr2[j]){
+ datensatz$END_JAHRx[k] <- as.numeric(imput_werte8$endjahr2[j])
+ print("Endjahr das imputiert wird")
+ print(datensatz$END_JAHRx[k])
+ # Indikator IMP_EJ auf 1 setzen, d.h. das Endjahr wurde imputiert
+ datensatz$IMP_EJ[k] <- 1
+ }
+ else{
+ if (datensatz$ANF_JAHR[k] < imput_werte8$endjahr3[j]){
+ datensatz$END_JAHRx[k] <- as.numeric(imput_werte8$endjahr3[j])
+ print("Endjahr das imputiert wird")
+ print(datensatz$END_JAHRx[k])
+ # Indikator IMP_EJ auf 1 setzen, d.h. das Endjahr wurde imputiert
+ datensatz$IMP_EJ[k] <- 1
+ }
+ else{
+ if (datensatz$ANF_JAHR[k] < imput_werte8$endjahr4[j]){
+ datensatz$END_JAHRx[k] <- as.numeric(imput_werte8$endjahr4[j])
+ print("Endjahr das imputiert wird")
+ print(datensatz$END_JAHRx[k])
+ # Indikator IMP_EJ auf 1 setzen, d.h. das Endjahr wurde imputiert
```

```
+ datensatz$IMP_EJ[k] <- 1
+ }
+ else{
+ if (datensatz$ANF_JAHR[k] < imput_werte8$endjahr5[j]){
+ datensatz$END_JAHRx[k] <- as.numeric(imput_werte8$endjahr5[j])
+ print("Endjahr das imputiert wird")
+ print(datensatz$END_JAHRx[k])
+ # Indikator IMP_EJ auf 1 setzen, d.h. das Endjahr wurde imputiert
+ datensatz$IMP_EJ[k] <- 1
+ }
+ else{
+ if (datensatz$ANF_JAHR[k] < imput_werte8$endjahr6[j]){
+ datensatz$END_JAHRx[k] <- as.numeric(imput_werte8$endjahr6[j])
+ print("Endjahr das imputiert wird")
+ print(datensatz$END_JAHRx[k])
+ # Indikator IMP_EJ auf 1 setzen, d.h. das Endjahr wurde imputiert
+ datensatz$IMP_EJ[k] <- 1
+ }
+ else{
+ if (datensatz$ANF_JAHR[k] < imput_werte8$endjahr7[j]){
+ datensatz$END_JAHRx[k] <- as.numeric(imput_werte8$endjahr7[j])
+ print("Endjahr das imputiert wird")
+ print(datensatz$END_JAHRx[k])
+ # Indikator IMP_EJ auf 1 setzen, d.h. das Endjahr wurde imputiert
+ datensatz$IMP_EJ[k] <- 1
+ }
+ else{
+ if (datensatz$ANF_JAHR[k] < imput_werte8$endjahr8[j]){
+ datensatz$END_JAHRx[k] <- as.numeric(imput_werte8$endjahr8[j])
+ print("Endjahr das imputiert wird")
+ print(datensatz$END_JAHRx[k])
+ # Indikator IMP_EJ auf 1 setzen, d.h. das Endjahr wurde imputiert
+ datensatz$IMP_EJ[k] <- 1
+ }
+ else{
+ if (datensatz$ANF_JAHR[k] < imput_werte8$endjahr9[j]){
+ datensatz$END_JAHRx[k] <- as.numeric(imput_werte8$endjahr9[j])
+ print("Endjahr das imputiert wird")
+ print(datensatz$END_JAHRx[k])
+ # Indikator IMP_EJ auf 1 setzen, d.h. das Endjahr wurde imputiert
+ datensatz$IMP_EJ[k] <- 1
+ }
+ else{
+ if (datensatz$ANF_JAHR[k] < imput_werte8$endjahr10[j]){
+ datensatz$END_JAHRx[k] <- as.numeric(imput_werte8$endjahr10[j])
+ print("Endjahr das imputiert wird")
+ print(datensatz$END_JAHRx[k])
+ # Indikator IMP_EJ auf 1 setzen, d.h. das Endjahr wurde imputiert
+ datensatz$IMP_EJ[k] <- 1
+ }
+ else{
+ if (datensatz$ANF_JAHR[k] < imput_werte8$endjahr11[j]){
+ datensatz$END_JAHRx[k] <- as.numeric(imput_werte8$endjahr11[j])
+ print("Endjahr das imputiert wird")
+ print(datensatz$END_JAHRx[k])
+ # Indikator IMP_EJ auf 1 setzen, d.h. das Endjahr wurde imputiert
+ datensatz$IMP_EJ[k] <- 1
+ }
+ else{
+ if (datensatz$ANF_JAHR[k] < imput_werte8$endjahr12[j]){
+ datensatz$END_JAHRx[k] <- as.numeric(imput_werte8$endjahr12[j])
+ print("Endjahr das imputiert wird")
+ print(datensatz$END_JAHRx[k])
+ # Indikator IMP_EJ auf 1 setzen, d.h. das Endjahr wurde imputiert
+ datensatz$IMP_EJ[k] <- 1
+ }
+ else{
+ if (datensatz$ANF_JAHR[k] < imput_werte8$endjahr13[j]){
+ datensatz$END_JAHRx[k] <- as.numeric(imput_werte8$endjahr13[j])
+ print("Endjahr das imputiert wird")
+ print(datensatz$END_JAHRx[k])
+ # Indikator IMP_EJ auf 1 setzen, d.h. das Endjahr wurde imputiert
+ datensatz$IMP_EJ[k] <- 1
```

```
+ }
+ else{
+ if (datensatz$ANF_JAHR[k] < imput_werte8$endjahr14[j]){
+ datensatz$END_JAHRx[k] <- as.numeric(imput_werte8$endjahr14[j])
+ print("Endjahr das imputiert wird")
+ print(datensatz$END_JAHRx[k])
+ # Indikator IMP_EJ auf 1 setzen, d.h. das Endjahr wurde imputiert
+ datensatz$IMP_EJ[k] <- 1
+ }
+ else{
+ if (datensatz$ANF_JAHR[k] < imput_werte8$endjahr15[j]){
+ datensatz$END_JAHRx[k] <- as.numeric(imput_werte8$endjahr15[j])
+ print("Endjahr das imputiert wird")
+ print(datensatz$END_JAHRx[k])
+ # Indikator IMP_EJ auf 1 setzen, d.h. das Endjahr wurde imputiert
+ datensatz$IMP_EJ[k] <- 1
+ }
+ else{
+ if (datensatz$ANF_JAHR[k] < imput_werte8$endjahr16[j]){
+ datensatz$END_JAHRx[k] <- as.numeric(imput_werte8$endjahr16[j])
+ print("Endjahr das imputiert wird")
+ print(datensatz$END_JAHRx[k])
+ # Indikator IMP_EJ auf 1 setzen, d.h. das Endjahr wurde imputiert
+ datensatz$IMP_EJ[k] <- 1
+ }
+ else{
+ if (datensatz$ANF_JAHR[k] < imput_werte8$endjahr17[j]){
+ datensatz$END_JAHRx[k] <- as.numeric(imput_werte8$endjahr17[j])
+ print("Endjahr das imputiert wird")
+ print(datensatz$END_JAHRx[k])
+ # Indikator IMP_EJ auf 1 setzen, d.h. das Endjahr wurde imputiert
+ datensatz$IMP_EJ[k] <- 1
+ }
+ else{
+ if (datensatz$ANF_JAHR[k] < imput_werte8$endjahr18[j]){
+ datensatz$END_JAHRx[k] <- as.numeric(imput_werte8$endjahr18[j])
+ print("Endjahr das imputiert wird")
+ print(datensatz$END_JAHRx[k])
+ # Indikator IMP_EJ auf 1 setzen, d.h. das Endjahr wurde imputiert
+ datensatz$IMP_EJ[k] <- 1
+ }
+ else{
+ if (datensatz$ANF_JAHR[k] < imput_werte8$endjahr19[j]){
+ datensatz$END_JAHRx[k] <- as.numeric(imput_werte8$endjahr19[j])
+ print("Endjahr das imputiert wird")
+ print(datensatz$END_JAHRx[k])
+ # Indikator IMP_EJ auf 1 setzen, d.h. das Endjahr wurde imputiert
+ datensatz$IMP_EJ[k] <- 1
+ }
+ else{
+ if (datensatz$ANF_JAHR[k] < imput_werte8$endjahr20[j]){
+ datensatz$END_JAHRx[k] <- as.numeric(imput_werte8$endjahr20[j])
+ print("Endjahr das imputiert wird")
+ print(datensatz$END_JAHRx[k])
+ # Indikator IMP_EJ auf 1 setzen, d.h. das Endjahr wurde imputiert
+ datensatz$IMP_EJ[k] <- 1
+ }
+ }}}}}}}}}}
+ }
+ }
+ }
+ }
+ }
+ }
+ }
+ }
+ if (is.na(datensatz$ANF_MONAT[k]) | is.na(datensatz$END_MONATx[k]) |
+ !is.na(datensatz$ANF_MONAT[k])&!is.na(datensatz$END_MONATx[k])
+ & (datensatz$ANF_MONAT[k] <= datensatz$END_MONATx[k])){
+ if (datensatz$ANF_JAHR[k] <= imput_werte8$endjahr1[j]){
+ datensatz$END_JAHRx[k] <- as.numeric(imput_werte8$endjahr1[j])
+ print("Endjahr das imputiert wird")
```

```
+ print(datensatz$END_JAHRx[k])
+ # Indikator IMP_EJ auf 1 setzen, d.h. das Endjahr wurde imputiert
+ datensatz$IMP_EJ[k] <- 1
+ }
+ else{
+ if (datensatz$ANF_JAHR[k] <= imput_werte8$endjahr2[j]){
+ datensatz$END_JAHRx[k] <- as.numeric(imput_werte8$endjahr2[j])
+ print("Endjahr das imputiert wird")
+ print(datensatz$END_JAHRx[k])
+ # Indikator IMP_EJ auf 1 setzen, d.h. das Endjahr wurde imputiert
+ datensatz$IMP_EJ[k] <- 1
+ }
+ else{
+ if (datensatz$ANF_JAHR[k] <= imput_werte8$endjahr3[j]){
+ datensatz$END_JAHRx[k] <- as.numeric(imput_werte8$endjahr3[j])
+ print("Endjahr das imputiert wird")
+ print(datensatz$END_JAHRx[k])
+ # Indikator IMP_EJ auf 1 setzen, d.h. das Endjahr wurde imputiert
+ datensatz$IMP_EJ[k] <- 1
+ }
+ else{
+ if (datensatz$ANF_JAHR[k] <= imput_werte8$endjahr4[j]){
+ datensatz$END_JAHRx[k] <- as.numeric(imput_werte8$endjahr4[j])
+ print("Endjahr das imputiert wird")
+ print(datensatz$END_JAHRx[k])
+ # Indikator IMP_EJ auf 1 setzen, d.h. das Endjahr wurde imputiert
+ datensatz$IMP_EJ[k] <- 1
+ }
+ else{
+ if (datensatz$ANF_JAHR[k] <= imput_werte8$endjahr5[j]){
+ datensatz$END_JAHRx[k] <- as.numeric(imput_werte8$endjahr5[j])
+ print("Endjahr das imputiert wird")
+ print(datensatz$END_JAHRx[k])
+ # Indikator IMP_EJ auf 1 setzen, d.h. das Endjahr wurde imputiert
+ datensatz$IMP_EJ[k] <- 1
+ }
+ else{
+ if (datensatz$ANF_JAHR[k] <= imput_werte8$endjahr6[j]){
+ datensatz$END_JAHRx[k] <- as.numeric(imput_werte8$endjahr6[j])
+ print("Endjahr das imputiert wird")
+ print(datensatz$END_JAHRx[k])
+ # Indikator IMP_EJ auf 1 setzen, d.h. das Endjahr wurde imputiert
+ datensatz$IMP_EJ[k] <- 1
+ }
+ else{
+ if (datensatz$ANF_JAHR[k] <= imput_werte8$endjahr7[j]){
+ datensatz$END_JAHRx[k] <- as.numeric(imput_werte8$endjahr7[j])
+ print("Endjahr das imputiert wird")
+ print(datensatz$END_JAHRx[k])
+ # Indikator IMP_EJ auf 1 setzen, d.h. das Endjahr wurde imputiert
+ datensatz$IMP_EJ[k] <- 1
+ }
+ else{
+ if (datensatz$ANF_JAHR[k] <= imput_werte8$endjahr8[j]){
+ datensatz$END_JAHRx[k] <- as.numeric(imput_werte8$endjahr8[j])
+ print("Endjahr das imputiert wird")
+ print(datensatz$END_JAHRx[k])
+ # Indikator IMP_EJ auf 1 setzen, d.h. das Endjahr wurde imputiert
+ datensatz$IMP_EJ[k] <- 1
+ }
+ else{
+ if (datensatz$ANF_JAHR[k] <= imput_werte8$endjahr9[j]){
+ datensatz$END_JAHRx[k] <- as.numeric(imput_werte8$endjahr9[j])
+ print("Endjahr das imputiert wird")
+ print(datensatz$END_JAHRx[k])
+ # Indikator IMP_EJ auf 1 setzen, d.h. das Endjahr wurde imputiert
+ datensatz$IMP_EJ[k] <- 1
+ }
+ else{
+ if (datensatz$ANF_JAHR[k] <= imput_werte8$endjahr10[j]){
+ datensatz$END_JAHRx[k] <- as.numeric(imput_werte8$endjahr10[j])
+ print("Endjahr das imputiert wird")
+ print(datensatz$END_JAHRx[k])
```

```
+ # Indikator IMP_EJ auf 1 setzen, d.h. das Endjahr wurde imputiert
+ datensatz$IMP_EJ[k] <- 1
+ }
+ else{
+ if (datensatz$ANF_JAHR[k] <= imput_werte8$endjahr11[j]){
+ datensatz$END_JAHRx[k] <- as.numeric(imput_werte8$endjahr11[j])
+ print("Endjahr das imputiert wird")
+ print(datensatz$END_JAHRx[k])
+ # Indikator IMP_EJ auf 1 setzen, d.h. das Endjahr wurde imputiert
+ datensatz$IMP_EJ[k] <- 1
+ }
+ else{
+ if (datensatz$ANF_JAHR[k] <= imput_werte8$endjahr12[j]){
+ datensatz$END_JAHRx[k] <- as.numeric(imput_werte8$endjahr12[j])
+ print("Endjahr das imputiert wird")
+ print(datensatz$END_JAHRx[k])
+ # Indikator IMP_EJ auf 1 setzen, d.h. das Endjahr wurde imputiert
+ datensatz$IMP_EJ[k] <- 1
+ }
+ else{
+ if (datensatz$ANF_JAHR[k] <= imput_werte8$endjahr13[j]){
+ datensatz$END_JAHRx[k] <- as.numeric(imput_werte8$endjahr13[j])
+ print("Endjahr das imputiert wird")
+ print(datensatz$END_JAHRx[k])
+ # Indikator IMP_EJ auf 1 setzen, d.h. das Endjahr wurde imputiert
+ datensatz$IMP_EJ[k] <- 1
+ }
+ else{
+ if (datensatz$ANF_JAHR[k] <= imput_werte8$endjahr14[j]){
+ datensatz$END_JAHRx[k] <- as.numeric(imput_werte8$endjahr14[j])
+ print("Endjahr das imputiert wird")
+ print(datensatz$END_JAHRx[k])
+ # Indikator IMP_EJ auf 1 setzen, d.h. das Endjahr wurde imputiert
+ datensatz$IMP_EJ[k] <- 1
+ }
+ else{
+ if (datensatz$ANF_JAHR[k] <= imput_werte8$endjahr15[j]){
+ datensatz$END_JAHRx[k] <- as.numeric(imput_werte8$endjahr15[j])
+ print("Endjahr das imputiert wird")
+ print(datensatz$END_JAHRx[k])
+ # Indikator IMP_EJ auf 1 setzen, d.h. das Endjahr wurde imputiert
+ datensatz$IMP_EJ[k] <- 1
+ }
+ else{
+ if (datensatz$ANF_JAHR[k] <= imput_werte8$endjahr16[j]){
+ datensatz$END_JAHRx[k] <- as.numeric(imput_werte8$endjahr16[j])
+ print("Endjahr das imputiert wird")
+ print(datensatz$END_JAHRx[k])
+ # Indikator IMP_EJ auf 1 setzen, d.h. das Endjahr wurde imputiert
+ datensatz$IMP_EJ[k] <- 1
+ }
+ else{
+ if (datensatz$ANF_JAHR[k] <= imput_werte8$endjahr17[j]){
+ datensatz$END_JAHRx[k] <- as.numeric(imput_werte8$endjahr17[j])
+ print("Endjahr das imputiert wird")
+ print(datensatz$END_JAHRx[k])
+ # Indikator IMP_EJ auf 1 setzen, d.h. das Endjahr wurde imputiert
+ datensatz$IMP_EJ[k] <- 1
+ }
+ else{
+ if (datensatz$ANF_JAHR[k] <= imput_werte8$endjahr18[j]){
+ datensatz$END_JAHRx[k] <- as.numeric(imput_werte8$endjahr18[j])
+ print("Endjahr das imputiert wird")
+ print(datensatz$END_JAHRx[k])
+ # Indikator IMP_EJ auf 1 setzen, d.h. das Endjahr wurde imputiert
+ datensatz$IMP_EJ[k] <- 1
+ }
+ else{
+ if (datensatz$ANF_JAHR[k] <= imput_werte8$endjahr19[j]){
+ datensatz$END_JAHRx[k] <- as.numeric(imput_werte8$endjahr19[j])
+ print("Endjahr das imputiert wird")
+ print(datensatz$END_JAHRx[k])
+ # Indikator IMP_EJ auf 1 setzen, d.h. das Endjahr wurde imputiert
```

```
+ datensatz$IMP_EJ[k] <- 1
+ }
+ else{
+ if (datensatz$ANF_JAHR[k] <= imput_werte8$endjahr20[j]){
+ datensatz$END_JAHRx[k] <- as.numeric(imput_werte8$endjahr20[j])
+ print("Endjahr das imputiert wird")
+ print(datensatz$END_JAHRx[k])
+ # Indikator IMP_EJ auf 1 setzen, d.h. das Endjahr wurde imputiert
+ datensatz$IMP_EJ[k] <- 1
+ }
+ }}}}}}}}}}
+ }
+ }
+ }
+ }
+ }
+ }
+ }
+ }
+ }
+ }
+ }
+ j <- j+1
+ }
+
+ ##############################################################################
+ ####                  Anfangsmonat (ANF_MONAT) imputieren                  ####
+ ##############################################################################
+
+ # Indikator, der angibt ob das ANF_MONAT imputiert wurde anlegen
+ datensatz$IMP_AM <- 0
+
+ ############################## SOLAR I ########################################
+ # in SOLAR I nur bedingen auf SES_r
+ print("**************Imputation des Anfangsmonats in SOLAR I***************")
+ a5 <- subset(datensatz_s1, is.na(ANF_MONAT) & kuenstl_geloescht == 1) # hier nur
+ # die Zeilen drin in denen das ANF_MONAT künstlich gelöscht wurden
+
+ # Datensatz anlegen, in den jeweils index der Zeile und imputiertes ANF_MONAT
+ # geschrieben werden (1.Spalte: index, 2.Spalte: anfangsmonat)
+ imput_werte5 <- data.frame(index=numeric(nrow(a5)),
+ anfmonat1 = numeric(nrow(a5)), anfmonat2 = numeric(nrow(a5)),
+ anfmonat3 = numeric(nrow(a5)), anfmonat4 = numeric(nrow(a5)),
+ anfmonat5 = numeric(nrow(a5)), anfmonat6 = numeric(nrow(a5)),
+ anfmonat7 = numeric(nrow(a5)), anfmonat8 = numeric(nrow(a5)),
+ anfmonat9 = numeric(nrow(a5)), anfmonat10 = numeric(nrow(a5)),
+ anfmonat11 = numeric(nrow(a5)), anfmonat12 = numeric(nrow(a5)),
+ anfmonat13 = numeric(nrow(a5)), anfmonat14 = numeric(nrow(a5)),
+ anfmonat15 = numeric(nrow(a5)), anfmonat16 = numeric(nrow(a5)),
+ anfmonat17 = numeric(nrow(a5)), anfmonat18 = numeric(nrow(a5)),
+ anfmonat19 = numeric(nrow(a5)), anfmonat20 = numeric(nrow(a5))
+ )
+
+ # j auf 1 setzen
+ j <- 1
+
+ # Startwert setzen (der der Funktion als Argument übergeben wurde)
+ set.seed(startwert)
+
+ for (i in 1: nrow(a5)){
+ print("Nächste Stelle i")
+ # SES und Index an der i-ten Stelle aus subset betrachten
+ ses <- a5$SES_r[i]
+ print("SES an der Stelle i")
+ print(ses)
+ index <- a5$index[i]
+ print("Index an der Stelle i")
+ print(index)
+ # aus großen Datensatz alle mit gleichem SES ziehen, durch
+ # IMP_AM == 0 werden diejenigen Fälle NICHT zur Berechnung der W.keiten
+ # verwendet, bei denen das ANF_MONAT bereits imputiert wurde
+ b <- subset(datensatz, SES_r==ses & !is.na(ANF_MONAT) & STUDIE == 1
+  & IMP_AM==0 )
```

```
+ print("Anzahl der Fälle mit gleichem SES")
+ print(nrow(b))
+ # Hier gibt es auf jeden Fall noch andere Fälle mit gleichem SES ! D.h.
+ # nrow(b) > 0 immer !
+ table1 <- table(b$ANF_MONAT)
+ print("Table der Anfangsmonate bedingt auf SES_r")
+ print(table1)
+ prob1 <- prop.table(as.array(table1))  # Wahrscheinlichkeiten für ANF_MONAT
+ # berechnen
+ print("Wahrscheinlichkeiten der Anfangsmonate bedingt auf SES_r")
+ print(prob1)
+ # dann aus den Anfangsmonaten mit diesen Wahrscheinlichkeiten ziehen
+ # Anfangsmonat für die Imputation ziehen mit der Funktion sample
+ imput <- as.list(sample(names(prob1), size = 20, replace=TRUE, prob = prob1))
+ # Im Datensatz imputieren, der die Daten aus SOLAR I und SOLAR II enthält
+ print("Anfangsmonat das an dieser Stelle imputiert werden soll")
+ print(imput)
+ # Index der Zeile (index) in 1.Spalte des Datensatzes imput_werte5 speichern
+ imput_werte5$index[j] <- index
+ # Anfangsmonat (imput) in 2.Spalte des Datensatzes imput_werte5 speichern
+ imput_werte5$anfmonat1[j] <- as.numeric(imput[[1]])
+ imput_werte5$anfmonat2[j] <- as.numeric(imput[[2]])
+ imput_werte5$anfmonat3[j] <- as.numeric(imput[[3]])
+ imput_werte5$anfmonat4[j] <- as.numeric(imput[[4]])
+ imput_werte5$anfmonat5[j] <- as.numeric(imput[[5]])
+ imput_werte5$anfmonat6[j] <- as.numeric(imput[[6]])
+ imput_werte5$anfmonat7[j] <- as.numeric(imput[[7]])
+ imput_werte5$anfmonat8[j] <- as.numeric(imput[[8]])
+ imput_werte5$anfmonat9[j] <- as.numeric(imput[[9]])
+ imput_werte5$anfmonat10[j] <- as.numeric(imput[[10]])
+ imput_werte5$anfmonat11[j] <- as.numeric(imput[[11]])
+ imput_werte5$anfmonat12[j] <- as.numeric(imput[[12]])
+ imput_werte5$anfmonat13[j] <- as.numeric(imput[[13]])
+ imput_werte5$anfmonat14[j] <- as.numeric(imput[[14]])
+ imput_werte5$anfmonat15[j] <- as.numeric(imput[[15]])
+ imput_werte5$anfmonat16[j] <- as.numeric(imput[[16]])
+ imput_werte5$anfmonat17[j] <- as.numeric(imput[[17]])
+ imput_werte5$anfmonat18[j] <- as.numeric(imput[[18]])
+ imput_werte5$anfmonat19[j] <- as.numeric(imput[[19]])
+ imput_werte5$anfmonat20[j] <- as.numeric(imput[[20]])
+ print("Matrix,1.Spalte:Index der Zeile,2.Spalte:Zu imputierendes Anfangsmonat")
+ print(imput_werte5)
+ # Jetzt den Wert im "großen" Datensatz imputieren
+ for(k in 1:nrow(datensatz)){
+ # Wenn der Index übereinstimmt
+ if(datensatz$index[k] == imput_werte5$index[j]){
+ # Zu imputierendes Anfangsmonat in der Zeile mit diesem Index imputieren
+ if (datensatz$ANF_JAHR[k] != datensatz$END_JAHRx[k] |
+ (datensatz$ANF_JAHR[k] == datensatz$END_JAHRx[k]
+ & is.na(datensatz$END_MONATx[k]))){
+ datensatz$ANF_MONAT[k] <- as.numeric(imput_werte5$anfmonat1[j])
+ print("Anfangsmonat das imputiert wird")
+ print(datensatz$ANF_MONAT[k])
+ # Indikator IMP_AM auf 1 setzen, d.h. das Anfangsmonat wurde imputiert
+ datensatz$IMP_AM[k] <- 1
+ }
+ if (datensatz$ANF_JAHR[k] == datensatz$END_JAHRx[k]){
+ if (imput_werte5$anfmonat1[j] <= datensatz$END_MONATx[k]
+ | is.na(datensatz$END_MONATx[k])){
+ datensatz$ANF_MONAT[k] <- as.numeric(imput_werte5$anfmonat1[j])
+ print("Anfangsmonat das imputiert wird")
+ print(datensatz$ANF_MONAT[k])
+ # Indikator IMP_AM auf 1 setzen, d.h. das Anfangsmonat wurde imputiert
+ datensatz$IMP_AM[k] <- 1
+ }
+ else{
+ if (imput_werte5$anfmonat2[j] <= datensatz$END_MONATx[k]){
+ datensatz$ANF_MONAT[k] <- as.numeric(imput_werte5$anfmonat2[j])
+ print("Anfangsmonat das imputiert wird")
+ print(datensatz$ANF_MONAT[k])
+ # Indikator IMP_AM auf 1 setzen, d.h. das Anfangsmonat wurde imputiert
+ datensatz$IMP_AM[k] <- 1
+ }
```

```
+ else{
+ if (imput_werte5$anfmonat3[j] <= datensatz$END_MONATx[k]){
+ datensatz$ANF_MONAT[k] <- as.numeric(imput_werte5$anfmonat3[j])
+ print("Anfangsmonat das imputiert wird")
+ print(datensatz$ANF_MONAT[k])
+ # Indikator IMP_AM auf 1 setzen, d.h. das Anfangsmonat wurde imputiert
+ datensatz$IMP_AM[k] <- 1
+ }
+ else{
+ if (imput_werte5$anfmonat4[j] <= datensatz$END_MONATx[k]){
+ datensatz$ANF_MONAT[k] <- as.numeric(imput_werte5$anfmonat4[j])
+ print("Anfangsmonat das imputiert wird")
+ print(datensatz$ANF_MONAT[k])
+ # Indikator IMP_AM auf 1 setzen, d.h. das Anfangsmonat wurde imputiert
+ datensatz$IMP_AM[k] <- 1
+ }
+ else{
+ if (imput_werte5$anfmonat5[j] <= datensatz$END_MONATx[k]){
+ datensatz$ANF_MONAT[k] <- as.numeric(imput_werte5$anfmonat5[j])
+ print("Anfangsmonat das imputiert wird")
+ print(datensatz$ANF_MONAT[k])
+ # Indikator IMP_AM auf 1 setzen, d.h. das Anfangsmonat wurde imputiert
+ datensatz$IMP_AM[k] <- 1
+ }
+ else{
+ if (imput_werte5$anfmonat6[j] <= datensatz$END_MONATx[k]){
+ datensatz$ANF_MONAT[k] <- as.numeric(imput_werte5$anfmonat6[j])
+ print("Anfangsmonat das imputiert wird")
+ print(datensatz$ANF_MONAT[k])
+ # Indikator IMP_AM auf 1 setzen, d.h. das Anfangsmonat wurde imputiert
+ datensatz$IMP_AM[k] <- 1
+ }
+ else{
+ if (imput_werte5$anfmonat7[j] <= datensatz$END_MONATx[k]){
+ datensatz$ANF_MONAT[k] <- as.numeric(imput_werte5$anfmonat7[j])
+ print("Anfangsmonat das imputiert wird")
+ print(datensatz$ANF_MONAT[k])
+ # Indikator IMP_AM auf 1 setzen, d.h. das Anfangsmonat wurde imputiert
+ datensatz$IMP_AM[k] <- 1
+ }
+ else{
+ if (imput_werte5$anfmonat8[j] <= datensatz$END_MONATx[k]){
+ datensatz$ANF_MONAT[k] <- as.numeric(imput_werte5$anfmonat8[j])
+ print("Anfangsmonat das imputiert wird")
+ print(datensatz$ANF_MONAT[k])
+ # Indikator IMP_AM auf 1 setzen, d.h. das Anfangsmonat wurde imputiert
+ datensatz$IMP_AM[k] <- 1
+ }
+ else{
+ if (imput_werte5$anfmonat9[j] <= datensatz$END_MONATx[k]){
+ datensatz$ANF_MONAT[k] <- as.numeric(imput_werte5$anfmonat9[j])
+ print("Anfangsmonat das imputiert wird")
+ print(datensatz$ANF_MONAT[k])
+ # Indikator IMP_AM auf 1 setzen, d.h. das Anfangsmonat wurde imputiert
+ datensatz$IMP_AM[k] <- 1
+ }
+ else{
+ if (imput_werte5$anfmonat10[j] <= datensatz$END_MONATx[k]){
+ datensatz$ANF_MONAT[k] <- as.numeric(imput_werte5$anfmonat10[j])
+ print("Anfangsmonat das imputiert wird")
+ print(datensatz$ANF_MONAT[k])
+ # Indikator IMP_AM auf 1 setzen, d.h. das Anfangsmonat wurde imputiert
+ datensatz$IMP_AM[k] <- 1
+ }
+ else{
+ if (imput_werte5$anfmonat11[j] <= datensatz$END_MONATx[k]){
+ datensatz$ANF_MONAT[k] <- as.numeric(imput_werte5$anfmonat11[j])
+ print("Anfangsmonat das imputiert wird")
+ print(datensatz$ANF_MONAT[k])
+ # Indikator IMP_AM auf 1 setzen, d.h. das Anfangsmonat wurde imputiert
+ datensatz$IMP_AM[k] <- 1
+ }
+ else{
```

```
+ if (imput_werte5$anfmonat12[j] <= datensatz$END_MONATx[k]){
+ datensatz$ANF_MONAT[k] <- as.numeric(imput_werte5$anfmonat12[j])
+ print("Anfangsmonat das imputiert wird")
+ print(datensatz$ANF_MONAT[k])
+ # Indikator IMP_AM auf 1 setzen, d.h. das Anfangsmonat wurde imputiert
+ datensatz$IMP_AM[k] <- 1
+ }
+ else{
+ if (imput_werte5$anfmonat13[j] <= datensatz$END_MONATx[k]){
+ datensatz$ANF_MONAT[k] <- as.numeric(imput_werte5$anfmonat13[j])
+ print("Anfangsmonat das imputiert wird")
+ print(datensatz$ANF_MONAT[k])
+ # Indikator IMP_AM auf 1 setzen, d.h. das Anfangsmonat wurde imputiert
+ datensatz$IMP_AM[k] <- 1
+ }
+ else{
+ if (imput_werte5$anfmonat14[j] <= datensatz$END_MONATx[k]){
+ datensatz$ANF_MONAT[k] <- as.numeric(imput_werte5$anfmonat14[j])
+ print("Anfangsmonat das imputiert wird")
+ print(datensatz$ANF_MONAT[k])
+ # Indikator IMP_AM auf 1 setzen, d.h. das Anfangsmonat wurde imputiert
+ datensatz$IMP_AM[k] <- 1
+ }
+ else{
+ if (imput_werte5$anfmonat15[j] <= datensatz$END_MONATx[k]){
+ datensatz$ANF_MONAT[k] <- as.numeric(imput_werte5$anfmonat15[j])
+ print("Anfangsmonat das imputiert wird")
+ print(datensatz$ANF_MONAT[k])
+ # Indikator IMP_AM auf 1 setzen, d.h. das Anfangsmonat wurde imputiert
+ datensatz$IMP_AM[k] <- 1
+ }
+ else{
+ if (imput_werte5$anfmonat16[j] <= datensatz$END_MONATx[k]){
+ datensatz$ANF_MONAT[k] <- as.numeric(imput_werte5$anfmonat16[j])
+ print("Anfangsmonat das imputiert wird")
+ print(datensatz$ANF_MONAT[k])
+ # Indikator IMP_AM auf 1 setzen, d.h. das Anfangsmonat wurde imputiert
+ datensatz$IMP_AM[k] <- 1
+ }
+ else{
+ if (imput_werte5$anfmonat17[j] <= datensatz$END_MONATx[k]){
+ datensatz$ANF_MONAT[k] <- as.numeric(imput_werte5$anfmonat17[j])
+ print("Anfangsmonat das imputiert wird")
+ print(datensatz$ANF_MONAT[k])
+ # Indikator IMP_AM auf 1 setzen, d.h. das Anfangsmonat wurde imputiert
+ datensatz$IMP_AM[k] <- 1
+ }
+ else{
+ if (imput_werte5$anfmonat18[j] <= datensatz$END_MONATx[k]){
+ datensatz$ANF_MONAT[k] <- as.numeric(imput_werte5$anfmonat18[j])
+ print("Anfangsmonat das imputiert wird")
+ print(datensatz$ANF_MONAT[k])
+ # Indikator IMP_AM auf 1 setzen, d.h. das Anfangsmonat wurde imputiert
+ datensatz$IMP_AM[k] <- 1
+ }
+ else{
+ if (imput_werte5$anfmonat19[j] <= datensatz$END_MONATx[k]){
+ datensatz$ANF_MONAT[k] <- as.numeric(imput_werte5$anfmonat19[j])
+ print("Anfangsmonat das imputiert wird")
+ print(datensatz$ANF_MONAT[k])
+ # Indikator IMP_AM auf 1 setzen, d.h. das Anfangsmonat wurde imputiert
+ datensatz$IMP_AM[k] <- 1
+ }
+ else{
+ if (imput_werte5$anfmonat20[j] <= datensatz$END_MONATx[k]){
+ datensatz$ANF_MONAT[k] <- as.numeric(imput_werte5$anfmonat20[j])
+ print("Anfangsmonat das imputiert wird")
+ print(datensatz$ANF_MONAT[k])
+ # Indikator IMP_AM auf 1 setzen, d.h. das Anfangsmonat wurde imputiert
+ datensatz$IMP_AM[k] <- 1
+ }
+ }}}}}}}}}}
+ }
```

```
+ }
+ }
+ }
+ }
+ }
+ }
+ }
+ }
+ }
+ }
+ }
+ j <- j+1
+ }
+
+ ############################## SOLAR II ####################################
+ # in SOLAR II bedingen auf SES_r und s2BERUF
+ print("**************Imputation des Anfangsmonats in SOLAR II**************")
+ a6 <- subset(datensatz_s2, is.na(ANF_MONAT) & kuenstl_geloescht == 1) # hier nur
+ # die Zeilen drin in denen das ANF_MONAT künstlich gelöscht wurden
+
+ # Datensatz anlegen, in den jeweils index der Zeile und imputiertes ANF_MONAT
+ # geschrieben werden (1.Spalte: index, 2.Spalte: anfangsmonat)
+ imput_werte6 <- data.frame(index=numeric(nrow(a6)),
+ anfmonat1 = numeric(nrow(a6)), anfmonat2 = numeric(nrow(a6)),
+ anfmonat3 = numeric(nrow(a6)), anfmonat4 = numeric(nrow(a6)),
+ anfmonat5 = numeric(nrow(a6)), anfmonat6 = numeric(nrow(a6)),
+ anfmonat7 = numeric(nrow(a6)), anfmonat8 = numeric(nrow(a6)),
+ anfmonat9 = numeric(nrow(a6)), anfmonat10 = numeric(nrow(a6)),
+ anfmonat11 = numeric(nrow(a6)), anfmonat12 = numeric(nrow(a6)),
+ anfmonat13 = numeric(nrow(a6)), anfmonat14 = numeric(nrow(a6)),
+ anfmonat15 = numeric(nrow(a6)), anfmonat16 = numeric(nrow(a6)),
+ anfmonat17 = numeric(nrow(a6)), anfmonat18 = numeric(nrow(a6)),
+ anfmonat19 = numeric(nrow(a6)), anfmonat20 = numeric(nrow(a6))
+ )
+
+ # j auf 1 setzen
+ j <- 1
+
+ # Startwert setzen (der der Funktion als Argument übergeben wurde)
+ set.seed(startwert)
+
+ for (i in 1: nrow(a6)){
+ print("Nächste Stelle i")
+ # SES und Index und Beruf an der i-ten Stelle aus subset betrachten
+ beruf <- a6$s2BERUF[i]
+ print("s2BERUF an der Stelle i")
+ print(beruf)
+ ses <- a6$SES_r[i]
+ print("SES an der Stelle i")
+ print(ses)
+ index <- a6$index[i]
+ print("Index an der Stelle i")
+ print(index)
+ # aus großen Datensatz alle mit gleichem SES ziehen, durch
+ # IMP_AM == 0 werden diejenigen Fälle NICHT zur Berechnung der W.keiten
+ # verwendet, bei denen das ANF_MONAT bereits imputiert wurde
+ b <- subset(datensatz, s2BERUF == beruf & SES_r==ses & !is.na(ANF_MONAT) & STUDIE == 2
+  & IMP_AM==0 )
+ print("Anzahl der Fälle mit gleichem SES")
+ print(nrow(b))
+
+ if(nrow(b)>0){ # Wenn es noch andere Fälle mit gleichem s2BERUF und SES_r gibt
+ print("b groesser als 0 also auf s2BERUF und SES_r bedingen")
+ table1 <- table(b$ANF_MONAT)
+ print("Table der Anfangsmonate bedingt auf SES_r und s2BERUF")
+ print(table1)
+ prob1 <- prop.table(as.array(table1))  # Wahrscheinlichkeiten für ANF_MONAT
+ # berechnen
+ print("Wahrscheinlichkeiten der Anfangsmonate bedingt auf SES_r und s2BERUF")
+ print(prob1)
+ }
+ if(nrow(b)==0 | (nrow(b)==1 & b$END_MONATx[1] < a6$ANF_MONAT[i]) |
+ (nrow(b)==2 & b$END_MONATx[1] < a6$ANF_MONAT[i]
```

```
+ & b$END_MONATx[2] < a6$ANF_MONAT[i])| (nrow(b)==3
+ & b$END_MONATx[1] < a6$ANF_MONAT[i] & b$END_MONATx[2] < a6$ANF_MONAT[i]
+ & b$END_MONATx[3] < a6$ANF_MONAT[i])){
+ # Wenn es sonst keinen Fall mit gleichem s2BERUF und
+ # gleichem SES_r gibt bei dem das ANF_MONAT fehlt (d.h. ANF_MONAT fehlt nur in
+ # diesem einen Fall mit diesem s2BERUF und diesem SES_r)
+ # oder es gibt nur 1(2/3) fälle und bei denen wäre dann das Anfangsmonat >
+ # Endmonat
+ print("b gleich 0 also nur auf SES_r bedingen")
+ # aus großem Datensatz alle mit gleichem SES_r ziehen, durch
+ # IMP_AM == 0 werden diejenigen Fälle NICHT zur Berechnung der W.keiten
+ # verwendet, bei denen das ANF_MONAT bereits imputiert wurde
+ c <- subset(datensatz, SES_r==ses & !is.na(ANF_MONAT) & STUDIE == 2
+ & IMP_AM==0 )
+ table1 <- table(c$ANF_MONAT)
+ print("Table der Anfangsmonate bedingt auf SES_r")
+ print(table1)
+ prob1 <- prop.table(as.array(table1))  # Wahrscheinlichkeiten für ANF_MONAT
+ # berechnen
+ print("Wahrscheinlichkeiten der Anfangsmonate bedingt auf Geschlecht")
+ print(prob1)
+ }
+ # dann aus den Anfangsmonaten mit diesen Wahrscheinlichkeiten ziehen
+ # Anfangsmonat für die Imputation ziehen mit der Funktion sample
+ imput <- as.list(sample(names(prob1), size = 20, replace=TRUE, prob = prob1))
+ # Im Datensatz imputieren, der die Daten aus SOLAR I und SOLAR II enthält
+ print("Anfangsmonat das an dieser Stelle imputiert werden soll")
+ print(imput)
+ # Index der Zeile (index) in 1.Spalte des Datensatzes imput_werte6 speichern
+ imput_werte6$index[j] <- index
+ # Anfangsmonat (imput) in 2.Spalte des Datensatzes imput_werte6 speichern
+ imput_werte6$anfmonat1[j] <- as.numeric(imput[[1]])
+ imput_werte6$anfmonat2[j] <- as.numeric(imput[[2]])
+ imput_werte6$anfmonat3[j] <- as.numeric(imput[[3]])
+ imput_werte6$anfmonat4[j] <- as.numeric(imput[[4]])
+ imput_werte6$anfmonat5[j] <- as.numeric(imput[[5]])
+ imput_werte6$anfmonat6[j] <- as.numeric(imput[[6]])
+ imput_werte6$anfmonat7[j] <- as.numeric(imput[[7]])
+ imput_werte6$anfmonat8[j] <- as.numeric(imput[[8]])
+ imput_werte6$anfmonat9[j] <- as.numeric(imput[[9]])
+ imput_werte6$anfmonat10[j] <- as.numeric(imput[[10]])
+ imput_werte6$anfmonat11[j] <- as.numeric(imput[[11]])
+ imput_werte6$anfmonat12[j] <- as.numeric(imput[[12]])
+ imput_werte6$anfmonat13[j] <- as.numeric(imput[[13]])
+ imput_werte6$anfmonat14[j] <- as.numeric(imput[[14]])
+ imput_werte6$anfmonat15[j] <- as.numeric(imput[[15]])
+ imput_werte6$anfmonat16[j] <- as.numeric(imput[[16]])
+ imput_werte6$anfmonat17[j] <- as.numeric(imput[[17]])
+ imput_werte6$anfmonat18[j] <- as.numeric(imput[[18]])
+ imput_werte6$anfmonat19[j] <- as.numeric(imput[[19]])
+ imput_werte6$anfmonat20[j] <- as.numeric(imput[[20]])
+ print("Matrix,1.Spalte:Index der Zeile,2.Spalte:Zu imputierendes Anfangsmonat")
+ print(imput_werte6)
+ # Jetzt den Wert im "großen" Datensatz imputieren
+ for(k in 1:nrow(datensatz)){
+ # Wenn der Index übereinstimmt
+ if(datensatz$index[k] == imput_werte6$index[j]){
+ # Zu imputierendes Anfangsmonat in der Zeile mit diesem Index imputieren
+ if (datensatz$ANF_JAHR[k] != datensatz$END_JAHRx[k] |
+ (datensatz$ANF_JAHR[k] == datensatz$END_JAHRx[k]
+ & is.na(datensatz$END_MONATx[k]))){
+ datensatz$ANF_MONAT[k] <- as.numeric(imput_werte6$anfmonat1[j])
+ print("Anfangsmonat das imputiert wird")
+ print(datensatz$ANF_MONAT[k])
+ # Indikator IMP_AM auf 1 setzen, d.h. das Anfangsmonat wurde imputiert
+ datensatz$IMP_AM[k] <- 1
+ }
+ if (datensatz$ANF_JAHR[k] == datensatz$END_JAHRx[k]){
+ if (imput_werte6$anfmonat1[j] <= datensatz$END_MONATx[k]
+ | is.na(datensatz$END_MONATx[k])){
+ datensatz$ANF_MONAT[k] <- as.numeric(imput_werte6$anfmonat1[j])
+ print("Anfangsmonat das imputiert wird")
+ print(datensatz$ANF_MONAT[k])
```

```
+ # Indikator IMP_AM auf 1 setzen, d.h. das Anfangsmonat wurde imputiert
+ datensatz$IMP_AM[k] <- 1
+ }
+ else{
+ if (imput_werte6$anfmonat2[j] <= datensatz$END_MONATx[k]){
+ datensatz$ANF_MONAT[k] <- as.numeric(imput_werte6$anfmonat2[j])
+ print("Anfangsmonat das imputiert wird")
+ print(datensatz$ANF_MONAT[k])
+ # Indikator IMP_AM auf 1 setzen, d.h. das Anfangsmonat wurde imputiert
+ datensatz$IMP_AM[k] <- 1
+ }
+ else{
+ if (imput_werte6$anfmonat3[j] <= datensatz$END_MONATx[k]){
+ datensatz$ANF_MONAT[k] <- as.numeric(imput_werte6$anfmonat3[j])
+ print("Anfangsmonat das imputiert wird")
+ print(datensatz$ANF_MONAT[k])
+ # Indikator IMP_AM auf 1 setzen, d.h. das Anfangsmonat wurde imputiert
+ datensatz$IMP_AM[k] <- 1
+ }
+ else{
+ if (imput_werte6$anfmonat4[j] <= datensatz$END_MONATx[k]){
+ datensatz$ANF_MONAT[k] <- as.numeric(imput_werte6$anfmonat4[j])
+ print("Anfangsmonat das imputiert wird")
+ print(datensatz$ANF_MONAT[k])
+ # Indikator IMP_AM auf 1 setzen, d.h. das Anfangsmonat wurde imputiert
+ datensatz$IMP_AM[k] <- 1
+ }
+ else{
+ if (imput_werte6$anfmonat5[j] <= datensatz$END_MONATx[k]){
+ datensatz$ANF_MONAT[k] <- as.numeric(imput_werte6$anfmonat5[j])
+ print("Anfangsmonat das imputiert wird")
+ print(datensatz$ANF_MONAT[k])
+ # Indikator IMP_AM auf 1 setzen, d.h. das Anfangsmonat wurde imputiert
+ datensatz$IMP_AM[k] <- 1
+ }
+ else{
+ if (imput_werte6$anfmonat6[j] <= datensatz$END_MONATx[k]){
+ datensatz$ANF_MONAT[k] <- as.numeric(imput_werte6$anfmonat6[j])
+ print("Anfangsmonat das imputiert wird")
+ print(datensatz$ANF_MONAT[k])
+ # Indikator IMP_AM auf 1 setzen, d.h. das Anfangsmonat wurde imputiert
+ datensatz$IMP_AM[k] <- 1
+ }
+ else{
+ if (imput_werte6$anfmonat7[j] <= datensatz$END_MONATx[k]){
+ datensatz$ANF_MONAT[k] <- as.numeric(imput_werte6$anfmonat7[j])
+ print("Anfangsmonat das imputiert wird")
+ print(datensatz$ANF_MONAT[k])
+ # Indikator IMP_AM auf 1 setzen, d.h. das Anfangsmonat wurde imputiert
+ datensatz$IMP_AM[k] <- 1
+ }
+ else{
+ if (imput_werte6$anfmonat8[j] <= datensatz$END_MONATx[k]){
+ datensatz$ANF_MONAT[k] <- as.numeric(imput_werte6$anfmonat8[j])
+ print("Anfangsmonat das imputiert wird")
+ print(datensatz$ANF_MONAT[k])
+ # Indikator IMP_AM auf 1 setzen, d.h. das Anfangsmonat wurde imputiert
+ datensatz$IMP_AM[k] <- 1
+ }
+ else{
+ if (imput_werte6$anfmonat9[j] <= datensatz$END_MONATx[k]){
+ datensatz$ANF_MONAT[k] <- as.numeric(imput_werte6$anfmonat9[j])
+ print("Anfangsmonat das imputiert wird")
+ print(datensatz$ANF_MONAT[k])
+ # Indikator IMP_AM auf 1 setzen, d.h. das Anfangsmonat wurde imputiert
+ datensatz$IMP_AM[k] <- 1
+ }
+ else{
+ if (imput_werte6$anfmonat10[j] <= datensatz$END_MONATx[k]){
+ datensatz$ANF_MONAT[k] <- as.numeric(imput_werte6$anfmonat10[j])
+ print("Anfangsmonat das imputiert wird")
+ print(datensatz$ANF_MONAT[k])
+ # Indikator IMP_AM auf 1 setzen, d.h. das Anfangsmonat wurde imputiert
```

```
+ datensatz$IMP_AM[k] <- 1
+ }
+ else{
+ if (imput_werte6$anfmonat11[j] <= datensatz$END_MONATx[k]){
+ datensatz$ANF_MONAT[k] <- as.numeric(imput_werte6$anfmonat11[j])
+ print("Anfangsmonat das imputiert wird")
+ print(datensatz$ANF_MONAT[k])
+ # Indikator IMP_AM auf 1 setzen, d.h. das Anfangsmonat wurde imputiert
+ datensatz$IMP_AM[k] <- 1
+ }
+ else{
+ if (imput_werte6$anfmonat12[j] <= datensatz$END_MONATx[k]){
+ datensatz$ANF_MONAT[k] <- as.numeric(imput_werte6$anfmonat12[j])
+ print("Anfangsmonat das imputiert wird")
+ print(datensatz$ANF_MONAT[k])
+ # Indikator IMP_AM auf 1 setzen, d.h. das Anfangsmonat wurde imputiert
+ datensatz$IMP_AM[k] <- 1
+ }
+ else{
+ if (imput_werte6$anfmonat13[j] <= datensatz$END_MONATx[k]){
+ datensatz$ANF_MONAT[k] <- as.numeric(imput_werte6$anfmonat13[j])
+ print("Anfangsmonat das imputiert wird")
+ print(datensatz$ANF_MONAT[k])
+ # Indikator IMP_AM auf 1 setzen, d.h. das Anfangsmonat wurde imputiert
+ datensatz$IMP_AM[k] <- 1
+ }
+ else{
+ if (imput_werte6$anfmonat14[j] <= datensatz$END_MONATx[k]){
+ datensatz$ANF_MONAT[k] <- as.numeric(imput_werte6$anfmonat14[j])
+ print("Anfangsmonat das imputiert wird")
+ print(datensatz$ANF_MONAT[k])
+ # Indikator IMP_AM auf 1 setzen, d.h. das Anfangsmonat wurde imputiert
+ datensatz$IMP_AM[k] <- 1
+ }
+ else{
+ if (imput_werte6$anfmonat15[j] <= datensatz$END_MONATx[k]){
+ datensatz$ANF_MONAT[k] <- as.numeric(imput_werte6$anfmonat15[j])
+ print("Anfangsmonat das imputiert wird")
+ print(datensatz$ANF_MONAT[k])
+ # Indikator IMP_AM auf 1 setzen, d.h. das Anfangsmonat wurde imputiert
+ datensatz$IMP_AM[k] <- 1
+ }
+ else{
+ if (imput_werte6$anfmonat16[j] <= datensatz$END_MONATx[k]){
+ datensatz$ANF_MONAT[k] <- as.numeric(imput_werte6$anfmonat16[j])
+ print("Anfangsmonat das imputiert wird")
+ print(datensatz$ANF_MONAT[k])
+ # Indikator IMP_AM auf 1 setzen, d.h. das Anfangsmonat wurde imputiert
+ datensatz$IMP_AM[k] <- 1
+ }
+ else{
+ if (imput_werte6$anfmonat17[j] <= datensatz$END_MONATx[k]){
+ datensatz$ANF_MONAT[k] <- as.numeric(imput_werte6$anfmonat17[j])
+ print("Anfangsmonat das imputiert wird")
+ print(datensatz$ANF_MONAT[k])
+ # Indikator IMP_AM auf 1 setzen, d.h. das Anfangsmonat wurde imputiert
+ datensatz$IMP_AM[k] <- 1
+ }
+ else{
+ if (imput_werte6$anfmonat18[j] <= datensatz$END_MONATx[k]){
+ datensatz$ANF_MONAT[k] <- as.numeric(imput_werte6$anfmonat18[j])
+ print("Anfangsmonat das imputiert wird")
+ print(datensatz$ANF_MONAT[k])
+ # Indikator IMP_AM auf 1 setzen, d.h. das Anfangsmonat wurde imputiert
+ datensatz$IMP_AM[k] <- 1
+ }
+ else{
+ if (imput_werte6$anfmonat19[j] <= datensatz$END_MONATx[k]){
+ datensatz$ANF_MONAT[k] <- as.numeric(imput_werte6$anfmonat19[j])
+ print("Anfangsmonat das imputiert wird")
+ print(datensatz$ANF_MONAT[k])
+ # Indikator IMP_AM auf 1 setzen, d.h. das Anfangsmonat wurde imputiert
+ datensatz$IMP_AM[k] <- 1
```

```
+ }
+ else{
+ if (imput_werte6$anfmonat20[j] <= datensatz$END_MONATx[k]){
+ datensatz$ANF_MONAT[k] <- as.numeric(imput_werte6$anfmonat20[j])
+ print("Anfangsmonat das imputiert wird")
+ print(datensatz$ANF_MONAT[k])
+ # Indikator IMP_AM auf 1 setzen, d.h. das Anfangsmonat wurde imputiert
+ datensatz$IMP_AM[k] <- 1
+ }
+ }}}}}}}}}
+ }
+ }
+ }
+ }
+ }
+ }
+ }
+ }
+ }
+ }
+ }
+ }
+ }
+ j <- j+1
+ }
+
+ ##############################################################################
+ ####                  Endmonat (END_MONATx) imputieren                    ####
+ ##############################################################################
+
+ # Indikator, der angibt ob das END_MONATx imputiert wurde anlegen
+ datensatz$IMP_EM <- 0
+
+ ################################ SOLAR I ######################################
+ # in SOLAR I nur bedingen auf SES_r
+ print("***************Imputation des Endmonats in SOLAR I***************")
+ a9 <- subset(datensatz_s1, is.na(END_MONATx) & kuenstl_geloescht == 1)# hier nur
+ # die Zeilen drin in denen das END_JAHRx künstlich gelöscht wurden
+
+ # Datensatz anlegen, in den jeweils index der Zeile und imputiertes END_MONATx
+ # geschrieben werden (1.Spalte: index, 2.Spalte: endmonat)
+ imput_werte9 <- data.frame(index=numeric(nrow(a9)),
+ endmonat1 = numeric(nrow(a9)), endmonat2 = numeric(nrow(a9)),
+ endmonat3 = numeric(nrow(a9)), endmonat4 = numeric(nrow(a9)),
+ endmonat5 = numeric(nrow(a9)), endmonat6 = numeric(nrow(a9)),
+ endmonat7 = numeric(nrow(a9)), endmonat8 = numeric(nrow(a9)),
+ endmonat9 = numeric(nrow(a9)), endmonat10 = numeric(nrow(a9)),
+ endmonat11 = numeric(nrow(a9)), endmonat12 = numeric(nrow(a9)),
+ endmonat13 = numeric(nrow(a9)), endmonat14 = numeric(nrow(a9)),
+ endmonat15 = numeric(nrow(a9)), endmonat16 = numeric(nrow(a9)),
+ endmonat17 = numeric(nrow(a9)), endmonat18 = numeric(nrow(a9)),
+ endmonat19 = numeric(nrow(a9)), endmonat20 = numeric(nrow(a9))
+ )
+
+ # j auf 1 setzen
+ j <- 1
+
+ # Startwert setzen (der der Funktion als Argument übergeben wurde)
+ set.seed(startwert)
+
+ for (i in 1: nrow(a9)){
+ print("Nächste Stelle i")
+ # SES und Index an der i-ten Stelle aus subset betrachten
+ ses <- a9$SES_r[i]
+ print("SES an der Stelle i")
+ print(ses)
+ index <- a9$index[i]
+ print("Index an der Stelle i")
+ print(index)
+ # aus großen Datensatz alle mit gleichem SES ziehen, durch
+ # IMP_EM == 0 werden diejenigen Fälle NICHT zur Berechnung der W.keiten
+ # verwendet, bei denen das END_MONATx bereits imputiert wurde
+ b <- subset(datensatz, SES_r==ses & !is.na(END_MONATx) & STUDIE == 1
```

```
+  & IMP_EM==0 )
+ print("Anzahl der Fälle mit gleichem SES")
+ print(nrow(b))
+ # Hier gibt es auf jeden Fall noch andere Fälle mit gleichem SES ! D.h.
+ # nrow(b) > 0 immer !
+ table1 <- table(b$END_MONATx)
+ print("Table der Endmonate bedingt auf SES_r")
+ print(table1)
+ prob1 <- prop.table(as.array(table1))  # Wahrscheinlichkeiten für END_MONATx
+ # berechnen
+ print("Wahrscheinlichkeiten der Endmonate bedingt auf SES_r")
+ print(prob1)
+ # dann aus den Endmonaten mit diesen Wahrscheinlichkeiten ziehen
+ # Endmonat für die Imputation ziehen mit der Funktion sample
+ imput <- as.list(sample(names(prob1), size = 20, replace=TRUE, prob = prob1))
+ # Im Datensatz imputieren, der die Daten aus SOLAR I und SOLAR II enthält
+ print("Endmonat das an dieser Stelle imputiert werden soll")
+ print(imput)
+ # Index der Zeile (index) in 1.Spalte des Datensatzes imput_werte7 speichern
+ imput_werte9$index[j] <- index
+ # Endmonat (imput) in 2.Spalte des Datensatzes imput_werte7 speichern
+ imput_werte9$endmonat1[j] <- as.numeric(imput[[1]])
+ imput_werte9$endmonat2[j] <- as.numeric(imput[[2]])
+ imput_werte9$endmonat3[j] <- as.numeric(imput[[3]])
+ imput_werte9$endmonat4[j] <- as.numeric(imput[[4]])
+ imput_werte9$endmonat5[j] <- as.numeric(imput[[5]])
+ imput_werte9$endmonat6[j] <- as.numeric(imput[[6]])
+ imput_werte9$endmonat7[j] <- as.numeric(imput[[7]])
+ imput_werte9$endmonat8[j] <- as.numeric(imput[[8]])
+ imput_werte9$endmonat9[j] <- as.numeric(imput[[9]])
+ imput_werte9$endmonat10[j] <- as.numeric(imput[[10]])
+ imput_werte9$endmonat11[j] <- as.numeric(imput[[11]])
+ imput_werte9$endmonat12[j] <- as.numeric(imput[[12]])
+ imput_werte9$endmonat13[j] <- as.numeric(imput[[13]])
+ imput_werte9$endmonat14[j] <- as.numeric(imput[[14]])
+ imput_werte9$endmonat15[j] <- as.numeric(imput[[15]])
+ imput_werte9$endmonat16[j] <- as.numeric(imput[[16]])
+ imput_werte9$endmonat17[j] <- as.numeric(imput[[17]])
+ imput_werte9$endmonat18[j] <- as.numeric(imput[[18]])
+ imput_werte9$endmonat19[j] <- as.numeric(imput[[19]])
+ imput_werte9$endmonat20[j] <- as.numeric(imput[[20]])
+ print("Matrix,1.Spalte:Index der Zeile,2.Spalte:Zu imputierendes Endmonat")
+ print(imput_werte9)
+ # Jetzt den Wert im "großen" Datensatz imputieren
+ for(k in 1:nrow(datensatz)){
+ # Wenn der Index übereinstimmt
+ if(datensatz$index[k] == imput_werte9$index[j]){
+ # Zu imputierendes Endmonat in der Zeile mit diesem Index imputieren
+ if (datensatz$ANF_JAHR[k] != datensatz$END_JAHRx[k]){
+ datensatz$END_MONATx[k] <- as.numeric(imput_werte9$endmonat1[j])
+ print("Endmonat das imputiert wird")
+ print(datensatz$END_MONATx[k])
+ # Indikator IMP_EM auf 1 setzen, d.h. das Endmonat wurde imputiert
+ datensatz$IMP_EM[k] <- 1
+ }
+ if (datensatz$ANF_JAHR[k] == datensatz$END_JAHRx[k]){
+ if (imput_werte9$endmonat1[j] >= datensatz$ANF_MONAT[k]){
+ datensatz$END_MONATx[k] <- as.numeric(imput_werte9$endmonat1[j])
+ print("Endmonat das imputiert wird")
+ print(datensatz$END_MONATx[k])
+ # Indikator IMP_EM auf 1 setzen, d.h. das Endmonat wurde imputiert
+ datensatz$IMP_EM[k] <- 1
+ }
+ else{
+ if (imput_werte9$endmonat2[j] >= datensatz$ANF_MONAT[k]){
+ datensatz$END_MONATx[k] <- as.numeric(imput_werte9$endmonat2[j])
+ print("Endmonat das imputiert wird")
+ print(datensatz$END_MONATx[k])
+ # Indikator IMP_EM auf 1 setzen, d.h. das Endmonat wurde imputiert
+ datensatz$IMP_EM[k] <- 1
+ }
+ else{
+ if (imput_werte9$endmonat3[j] >= datensatz$ANF_MONAT[k]){
```

```
+ datensatz$END_MONATx[k] <- as.numeric(imput_werte9$endmonat3[j])
+ print("Endmonat das imputiert wird")
+ print(datensatz$END_MONATx[k])
+ # Indikator IMP_EM auf 1 setzen, d.h. das Endmonat wurde imputiert
+ datensatz$IMP_EM[k] <- 1
+ }
+ else{
+ if (imput_werte9$endmonat4[j] >= datensatz$ANF_MONAT[k]){
+ datensatz$END_MONATx[k] <- as.numeric(imput_werte9$endmonat4[j])
+ print("Endmonat das imputiert wird")
+ print(datensatz$END_MONATx[k])
+ # Indikator IMP_EM auf 1 setzen, d.h. das Endmonat wurde imputiert
+ datensatz$IMP_EM[k] <- 1
+ }
+ else{
+ if (imput_werte9$endmonat5[j] >= datensatz$ANF_MONAT[k]){
+ datensatz$END_MONATx[k] <- as.numeric(imput_werte9$endmonat5[j])
+ print("Endmonat das imputiert wird")
+ print(datensatz$END_MONATx[k])
+ # Indikator IMP_EM auf 1 setzen, d.h. das Endmonat wurde imputiert
+ datensatz$IMP_EM[k] <- 1
+ }
+ else{
+ if (imput_werte9$endmonat6[j] >= datensatz$ANF_MONAT[k]){
+ datensatz$END_MONATx[k] <- as.numeric(imput_werte9$endmonat6[j])
+ print("Endmonat das imputiert wird")
+ print(datensatz$END_MONATx[k])
+ # Indikator IMP_EM auf 1 setzen, d.h. das Endmonat wurde imputiert
+ datensatz$IMP_EM[k] <- 1
+ }
+ else{
+ if (imput_werte9$endmonat7[j] >= datensatz$ANF_MONAT[k]){
+ datensatz$END_MONATx[k] <- as.numeric(imput_werte9$endmonat7[j])
+ print("Endmonat das imputiert wird")
+ print(datensatz$END_MONATx[k])
+ # Indikator IMP_EM auf 1 setzen, d.h. das Endmonat wurde imputiert
+ datensatz$IMP_EM[k] <- 1
+ }
+ else{
+ if (imput_werte9$endmonat8[j] >= datensatz$ANF_MONAT[k]){
+ datensatz$END_MONATx[k] <- as.numeric(imput_werte9$endmonat8[j])
+ print("Endmonat das imputiert wird")
+ print(datensatz$END_MONATx[k])
+ # Indikator IMP_EM auf 1 setzen, d.h. das Endmonat wurde imputiert
+ datensatz$IMP_EM[k] <- 1
+ }
+ else{
+ if (imput_werte9$endmonat9[j] >= datensatz$ANF_MONAT[k]){
+ datensatz$END_MONATx[k] <- as.numeric(imput_werte9$endmonat9[j])
+ print("Endmonat das imputiert wird")
+ print(datensatz$END_MONATx[k])
+ # Indikator IMP_EM auf 1 setzen, d.h. das Endmonat wurde imputiert
+ datensatz$IMP_EM[k] <- 1
+ }
+ else{
+ if (imput_werte9$endmonat10[j] >= datensatz$ANF_MONAT[k]){
+ datensatz$END_MONATx[k] <- as.numeric(imput_werte9$endmonat10[j])
+ print("Endmonat das imputiert wird")
+ print(datensatz$END_MONATx[k])
+ # Indikator IMP_EM auf 1 setzen, d.h. das Endmonat wurde imputiert
+ datensatz$IMP_EM[k] <- 1
+ }
+ else{
+ if (imput_werte9$endmonat11[j] >= datensatz$ANF_MONAT[k]){
+ datensatz$END_MONATx[k] <- as.numeric(imput_werte9$endmonat11[j])
+ print("Endmonat das imputiert wird")
+ print(datensatz$END_MONATx[k])
+ # Indikator IMP_EM auf 1 setzen, d.h. das Endmonat wurde imputiert
+ datensatz$IMP_EM[k] <- 1
+ }
+ else{
+ if (imput_werte9$endmonat12[j] >= datensatz$ANF_MONAT[k]){
+ datensatz$END_MONATx[k] <- as.numeric(imput_werte9$endmonat12[j])
```

```
+ print("Endmonat das imputiert wird")
+ print(datensatz$END_MONATx[k])
+ # Indikator IMP_EM auf 1 setzen, d.h. das Endmonat wurde imputiert
+ datensatz$IMP_EM[k] <- 1
+ }
+ else{
+ if (imput_werte9$endmonat13[j] >= datensatz$ANF_MONAT[k]){
+ datensatz$END_MONATx[k] <- as.numeric(imput_werte9$endmonat13[j])
+ print("Endmonat das imputiert wird")
+ print(datensatz$END_MONATx[k])
+ # Indikator IMP_EM auf 1 setzen, d.h. das Endmonat wurde imputiert
+ datensatz$IMP_EM[k] <- 1
+ }
+ else{
+ if (imput_werte9$endmonat14[j] >= datensatz$ANF_MONAT[k]){
+ datensatz$END_MONATx[k] <- as.numeric(imput_werte9$endmonat14[j])
+ print("Endmonat das imputiert wird")
+ print(datensatz$END_MONATx[k])
+ # Indikator IMP_EM auf 1 setzen, d.h. das Endmonat wurde imputiert
+ datensatz$IMP_EM[k] <- 1
+ }
+ else{
+ if (imput_werte9$endmonat15[j] >= datensatz$ANF_MONAT[k]){
+ datensatz$END_MONATx[k] <- as.numeric(imput_werte9$endmonat15[j])
+ print("Endmonat das imputiert wird")
+ print(datensatz$END_MONATx[k])
+ # Indikator IMP_EM auf 1 setzen, d.h. das Endmonat wurde imputiert
+ datensatz$IMP_EM[k] <- 1
+ }
+ else{
+ if (imput_werte9$endmonat16[j] >= datensatz$ANF_MONAT[k]){
+ datensatz$END_MONATx[k] <- as.numeric(imput_werte9$endmonat16[j])
+ print("Endmonat das imputiert wird")
+ print(datensatz$END_MONATx[k])
+ # Indikator IMP_EM auf 1 setzen, d.h. das Endmonat wurde imputiert
+ datensatz$IMP_EM[k] <- 1
+ }
+ else{
+ if (imput_werte9$endmonat17[j] >= datensatz$ANF_MONAT[k]){
+ datensatz$END_MONATx[k] <- as.numeric(imput_werte9$endmonat17[j])
+ print("Endmonat das imputiert wird")
+ print(datensatz$END_MONATx[k])
+ # Indikator IMP_EM auf 1 setzen, d.h. das Endmonat wurde imputiert
+ datensatz$IMP_EM[k] <- 1
+ }
+ else{
+ if (imput_werte9$endmonat18[j] >= datensatz$ANF_MONAT[k]){
+ datensatz$END_MONATx[k] <- as.numeric(imput_werte9$endmonat18[j])
+ print("Endmonat das imputiert wird")
+ print(datensatz$END_MONATx[k])
+ # Indikator IMP_EM auf 1 setzen, d.h. das Endmonat wurde imputiert
+ datensatz$IMP_EM[k] <- 1
+ }
+ else{
+ if (imput_werte9$endmonat19[j] >= datensatz$ANF_MONAT[k]){
+ datensatz$END_MONATx[k] <- as.numeric(imput_werte9$endmonat19[j])
+ print("Endmonat das imputiert wird")
+ print(datensatz$END_MONATx[k])
+ # Indikator IMP_EM auf 1 setzen, d.h. das Endmonat wurde imputiert
+ datensatz$IMP_EM[k] <- 1
+ }
+ else{
+ if (imput_werte9$endmonat20[j] >= datensatz$ANF_MONAT[k]){
+ datensatz$END_MONATx[k] <- as.numeric(imput_werte9$endmonat20[j])
+ print("Endmonat das imputiert wird")
+ print(datensatz$END_MONATx[k])
+ # Indikator IMP_EM auf 1 setzen, d.h. das Endmonat wurde imputiert
+ datensatz$IMP_EM[k] <- 1
+ }
+ }}}}}}}}}
+ }
+ }
+ }
```

```
+ }
+ }
+ }
+ }
+ }
+ }
+ }
+ }
+ }
+ }
+ j <- j+1
+ }
+
+ ############################## SOLAR II #####################################
+ # in SOLAR II bedingen auf SES_r und s2BERUF
+ print("***************Imputation des Endmonats in SOLAR II***************")
+ a10 <- subset(datensatz_s2, is.na(END_MONATx) & kuenstl_geloescht == 1)
+ # hier nur die Zeilen drin in denen das END_MONATx künstlich gelöscht wurden
+
+ # Datensatz anlegen, in den jeweils index der Zeile und imputiertes END_MONATx
+ # geschrieben werden (1.Spalte: index, 2.Spalte: wochenstunden)
+ imput_werte10 <- data.frame(index=numeric(nrow(a10)),
+ endmonat1 = numeric(nrow(a10)), endmonat2 = numeric(nrow(a10)),
+ endmonat3 = numeric(nrow(a10)), endmonat4 = numeric(nrow(a10)),
+ endmonat5 = numeric(nrow(a10)), endmonat6 = numeric(nrow(a10)),
+ endmonat7 = numeric(nrow(a10)), endmonat8 = numeric(nrow(a10)),
+ endmonat9 = numeric(nrow(a10)), endmonat10 = numeric(nrow(a10)),
+ endmonat11 = numeric(nrow(a10)), endmonat12 = numeric(nrow(a10)),
+ endmonat13 = numeric(nrow(a10)), endmonat14 = numeric(nrow(a10)),
+ endmonat15 = numeric(nrow(a10)), endmonat16 = numeric(nrow(a10)),
+ endmonat17 = numeric(nrow(a10)), endmonat18 = numeric(nrow(a10)),
+ endmonat19 = numeric(nrow(a10)), endmonat20 = numeric(nrow(a10))
+ )
+
+ # j auf 1 setzen
+ j <- 1
+
+ # Startwert setzen (der der Funktion als Argument übergeben wurde)
+ set.seed(startwert)
+
+ for (i in 1: nrow(a10)){
+ print("Nächste Stelle i")
+ # SES und Index an der i-ten Stelle aus subset betrachten
+ beruf <- a10$s2BERUF[i]
+ print("s2BERUF an der Stelle i")
+ print(beruf)
+ ses <- a10$SES_r[i]
+ print("SES an der Stelle i")
+ print(ses)
+ index <- a10$index[i]
+ print("Index an der Stelle i")
+ print(index)
+ # aus großen Datensatz alle mit gleichem SES ziehen, durch
+ # IMP_EM == 0 werden diejenigen Fälle NICHT zur Berechnung der W.keiten
+ # verwendet, bei denen das END_MONATx bereits imputiert wurde
+ b <- subset(datensatz, s2BERUF == beruf & SES_r==ses & !is.na(END_MONATx) & STUDIE == 2
+  & IMP_EM==0 )
+ print("Anzahl der Fälle mit gleichem SES")
+ print(nrow(b))
+
+
+ if(nrow(b)>0){ # Wenn es noch andere Fälle mit gleichem s2BERUF und SES_r gibt
+ print("b groesser als 0 also auf s2BERUF und SES_r bedingen")
+ table1 <- table(b$END_MONATx)
+ print("Table der Endmonate bedingt auf SES_r und s2BERUF")
+ print(table1)
+ prob1 <- prop.table(as.array(table1))  # Wahrscheinlichkeiten für END_MONATx
+ # berechnen
+ print("Wahrscheinlichkeiten der Endmonate bedingt auf SES_r und s2BERUF")
+ print(prob1)
+ }
+ if(nrow(b)==0 | (nrow(b)==1 & b$END_MONATx[1] < a10$ANF_MONAT[i]) |
+ (nrow(b)==2 & b$END_MONATx[1] < a10$ANF_MONAT[i]
```

```
+ & b$END_MONATx[2] < a10$ANF_MONAT[i])| (nrow(b)==3
+ & b$END_MONATx[1] < a10$ANF_MONAT[i] & b$END_MONATx[2] < a10$ANF_MONAT[i]
+ & b$END_MONATx[3] < a10$ANF_MONAT[i])){
+ # Wenn es sonst keinen Fall mit gleichem s2BERUF und
+ # gleichem SES_r gibt bei dem das END_MONATx fehlt (d.h. END_MONATx fehlt nur in
+ # diesem einen Fall mit diesem s2BERUF und diesem SES_r)
+ # oder es gibt nur 1(2/3) fälle und bei denen wäre dann das Anfangsmonat >
+ # Endmonat
+ print("b gleich 0 also nur auf SES_r bedingen")
+ # aus großem Datensatz alle mit gleichem SES_r ziehen, durch
+ # IMP_EM == 0 werden diejenigen Fälle NICHT zur Berechnung der W.keiten
+ # verwendet, bei denen das END_MONAT bereits imputiert wurde
+ c <- subset(datensatz, SES_r==ses & !is.na(END_MONATx) & STUDIE == 2
+ & IMP_EM==0 )
+ table1 <- table(c$END_MONATx)
+ print("Table der Endmonate bedingt auf SES_r")
+ print(table1)
+ prob1 <- prop.table(as.array(table1))  # Wahrscheinlichkeiten für END_MONATx
+ # berechnen
+ print("Wahrscheinlichkeiten der Endmonate bedingt auf Geschlecht")
+ print(prob1)
+ }
+
+ # dann aus den Endmonaten mit diesen Wahrscheinlichkeiten ziehen
+ # Endmonat für die Imputation ziehen mit der Funktion sample
+ imput <- as.list(sample(names(prob1), size = 20, replace=TRUE, prob = prob1))
+ # Im Datensatz imputieren, der die Daten aus SOLAR I und SOLAR II enthält
+ print("Endmonat das an dieser Stelle imputiert werden soll")
+ print(imput)
+ # Index der Zeile (index) in 1.Spalte des Datensatzes imput_werte10 speichern
+ imput_werte10$index[j] <- index
+ # Endmonat (imput) in 2.Spalte des Datensatzes imput_werte10 speichern
+ imput_werte10$endmonat1[j] <- as.numeric(imput[[1]])
+ imput_werte10$endmonat2[j] <- as.numeric(imput[[2]])
+ imput_werte10$endmonat3[j] <- as.numeric(imput[[3]])
+ imput_werte10$endmonat4[j] <- as.numeric(imput[[4]])
+ imput_werte10$endmonat5[j] <- as.numeric(imput[[5]])
+ imput_werte10$endmonat6[j] <- as.numeric(imput[[6]])
+ imput_werte10$endmonat7[j] <- as.numeric(imput[[7]])
+ imput_werte10$endmonat8[j] <- as.numeric(imput[[8]])
+ imput_werte10$endmonat9[j] <- as.numeric(imput[[9]])
+ imput_werte10$endmonat10[j] <- as.numeric(imput[[10]])
+ imput_werte10$endmonat11[j] <- as.numeric(imput[[11]])
+ imput_werte10$endmonat12[j] <- as.numeric(imput[[12]])
+ imput_werte10$endmonat13[j] <- as.numeric(imput[[13]])
+ imput_werte10$endmonat14[j] <- as.numeric(imput[[14]])
+ imput_werte10$endmonat15[j] <- as.numeric(imput[[15]])
+ imput_werte10$endmonat16[j] <- as.numeric(imput[[16]])
+ imput_werte10$endmonat17[j] <- as.numeric(imput[[17]])
+ imput_werte10$endmonat18[j] <- as.numeric(imput[[18]])
+ imput_werte10$endmonat19[j] <- as.numeric(imput[[19]])
+ imput_werte10$endmonat20[j] <- as.numeric(imput[[20]])
+ print("Matrix,1.Spalte:Index der Zeile,2.Spalte:Zu imputierendes Endmonat")
+ print(imput_werte10)
+ # Jetzt den Wert im "großen" Datensatz imputieren
+ for(k in 1:nrow(datensatz)){
+ # Wenn der Index übereinstimmt
+ if(datensatz$index[k] == imput_werte10$index[j]){
+ # Zu imputierendes Endmonat in der Zeile mit diesem Index imputieren
+ if (datensatz$ANF_JAHR[k] != datensatz$END_JAHRx[k]){
+ datensatz$END_MONATx[k] <- as.numeric(imput_werte10$endmonat1[j])
+ print("Endmonat das imputiert wird")
+ print(datensatz$END_MONATx[k])
+ # Indikator IMP_EM auf 1 setzen, d.h. das Endmonat wurde imputiert
+ datensatz$IMP_EM[k] <- 1
+ }
+ if (datensatz$ANF_JAHR[k] == datensatz$END_JAHRx[k]){
+ if (imput_werte10$endmonat1[j] >= datensatz$ANF_MONAT[k]){
+ datensatz$END_MONATx[k] <- as.numeric(imput_werte10$endmonat1[j])
+ print("Endmonat das imputiert wird")
+ print(datensatz$END_MONATx[k])
+ # Indikator IMP_EM auf 1 setzen, d.h. das Endmonat wurde imputiert
+ datensatz$IMP_EM[k] <- 1
```

```
+ }
+ else{
+ if (imput_werte10$endmonat2[j] >= datensatz$ANF_MONAT[k]){
+ datensatz$END_MONATx[k] <- as.numeric(imput_werte10$endmonat2[j])
+ print("Endmonat das imputiert wird")
+ print(datensatz$END_MONATx[k])
+ # Indikator IMP_EM auf 1 setzen, d.h. das Endmonat wurde imputiert
+ datensatz$IMP_EM[k] <- 1
+ }
+ else{
+ if (imput_werte10$endmonat3[j] >= datensatz$ANF_MONAT[k]){
+ datensatz$END_MONATx[k] <- as.numeric(imput_werte10$endmonat3[j])
+ print("Endmonat das imputiert wird")
+ print(datensatz$END_MONATx[k])
+ # Indikator IMP_EM auf 1 setzen, d.h. das Endmonat wurde imputiert
+ datensatz$IMP_EM[k] <- 1
+ }
+ else{
+ if (imput_werte10$endmonat4[j] >= datensatz$ANF_MONAT[k]){
+ datensatz$END_MONATx[k] <- as.numeric(imput_werte10$endmonat4[j])
+ print("Endmonat das imputiert wird")
+ print(datensatz$END_MONATx[k])
+ # Indikator IMP_EM auf 1 setzen, d.h. das Endmonat wurde imputiert
+ datensatz$IMP_EM[k] <- 1
+ }
+ else{
+ if (imput_werte10$endmonat5[j] >= datensatz$ANF_MONAT[k]){
+ datensatz$END_MONATx[k] <- as.numeric(imput_werte10$endmonat5[j])
+ print("Endmonat das imputiert wird")
+ print(datensatz$END_MONATx[k])
+ # Indikator IMP_EM auf 1 setzen, d.h. das Endmonat wurde imputiert
+ datensatz$IMP_EM[k] <- 1
+ }
+ else{
+ if (imput_werte10$endmonat6[j] >= datensatz$ANF_MONAT[k]){
+ datensatz$END_MONATx[k] <- as.numeric(imput_werte10$endmonat6[j])
+ print("Endmonat das imputiert wird")
+ print(datensatz$END_MONATx[k])
+ # Indikator IMP_EM auf 1 setzen, d.h. das Endmonat wurde imputiert
+ datensatz$IMP_EM[k] <- 1
+ }
+ else{
+ if (imput_werte10$endmonat7[j] >= datensatz$ANF_MONAT[k]){
+ datensatz$END_MONATx[k] <- as.numeric(imput_werte10$endmonat7[j])
+ print("Endmonat das imputiert wird")
+ print(datensatz$END_MONATx[k])
+ # Indikator IMP_EM auf 1 setzen, d.h. das Endmonat wurde imputiert
+ datensatz$IMP_EM[k] <- 1
+ }
+ else{
+ if (imput_werte10$endmonat8[j] >= datensatz$ANF_MONAT[k]){
+ datensatz$END_MONATx[k] <- as.numeric(imput_werte10$endmonat8[j])
+ print("Endmonat das imputiert wird")
+ print(datensatz$END_MONATx[k])
+ # Indikator IMP_EM auf 1 setzen, d.h. das Endmonat wurde imputiert
+ datensatz$IMP_EM[k] <- 1
+ }
+ else{
+ if (imput_werte10$endmonat9[j] >= datensatz$ANF_MONAT[k]){
+ datensatz$END_MONATx[k] <- as.numeric(imput_werte10$endmonat9[j])
+ print("Endmonat das imputiert wird")
+ print(datensatz$END_MONATx[k])
+ # Indikator IMP_EM auf 1 setzen, d.h. das Endmonat wurde imputiert
+ datensatz$IMP_EM[k] <- 1
+ }
+ else{
+ if (imput_werte10$endmonat10[j] >= datensatz$ANF_MONAT[k]){
+ datensatz$END_MONATx[k] <- as.numeric(imput_werte10$endmonat10[j])
+ print("Endmonat das imputiert wird")
+ print(datensatz$END_MONATx[k])
+ # Indikator IMP_EM auf 1 setzen, d.h. das Endmonat wurde imputiert
+ datensatz$IMP_EM[k] <- 1
+ }
```

```
+ else{
+ if (imput_werte10$endmonat11[j] >= datensatz$ANF_MONAT[k]){
+ datensatz$END_MONATx[k] <- as.numeric(imput_werte10$endmonat11[j])
+ print("Endmonat das imputiert wird")
+ print(datensatz$END_MONATx[k])
+ # Indikator IMP_EM auf 1 setzen, d.h. das Endmonat wurde imputiert
+ datensatz$IMP_EM[k] <- 1
+ }
+ else{
+ if (imput_werte10$endmonat12[j] >= datensatz$ANF_MONAT[k]){
+ datensatz$END_MONATx[k] <- as.numeric(imput_werte10$endmonat12[j])
+ print("Endmonat das imputiert wird")
+ print(datensatz$END_MONATx[k])
+ # Indikator IMP_EM auf 1 setzen, d.h. das Endmonat wurde imputiert
+ datensatz$IMP_EM[k] <- 1
+ }
+ else{
+ if (imput_werte10$endmonat13[j] >= datensatz$ANF_MONAT[k]){
+ datensatz$END_MONATx[k] <- as.numeric(imput_werte10$endmonat13[j])
+ print("Endmonat das imputiert wird")
+ print(datensatz$END_MONATx[k])
+ # Indikator IMP_EM auf 1 setzen, d.h. das Endmonat wurde imputiert
+ datensatz$IMP_EM[k] <- 1
+ }
+ else{
+ if (imput_werte10$endmonat14[j] >= datensatz$ANF_MONAT[k]){
+ datensatz$END_MONATx[k] <- as.numeric(imput_werte10$endmonat14[j])
+ print("Endmonat das imputiert wird")
+ print(datensatz$END_MONATx[k])
+ # Indikator IMP_EM auf 1 setzen, d.h. das Endmonat wurde imputiert
+ datensatz$IMP_EM[k] <- 1
+ }
+ else{
+ if (imput_werte10$endmonat15[j] >= datensatz$ANF_MONAT[k]){
+ datensatz$END_MONATx[k] <- as.numeric(imput_werte10$endmonat15[j])
+ print("Endmonat das imputiert wird")
+ print(datensatz$END_MONATx[k])
+ # Indikator IMP_EM auf 1 setzen, d.h. das Endmonat wurde imputiert
+ datensatz$IMP_EM[k] <- 1
+ }
+ else{
+ if (imput_werte10$endmonat16[j] >= datensatz$ANF_MONAT[k]){
+ datensatz$END_MONATx[k] <- as.numeric(imput_werte10$endmonat16[j])
+ print("Endmonat das imputiert wird")
+ print(datensatz$END_MONATx[k])
+ # Indikator IMP_EM auf 1 setzen, d.h. das Endmonat wurde imputiert
+ datensatz$IMP_EM[k] <- 1
+ }
+ else{
+ if (imput_werte10$endmonat17[j] >= datensatz$ANF_MONAT[k]){
+ datensatz$END_MONATx[k] <- as.numeric(imput_werte10$endmonat17[j])
+ print("Endmonat das imputiert wird")
+ print(datensatz$END_MONATx[k])
+ # Indikator IMP_EM auf 1 setzen, d.h. das Endmonat wurde imputiert
+ datensatz$IMP_EM[k] <- 1
+ }
+ else{
+ if (imput_werte10$endmonat18[j] >= datensatz$ANF_MONAT[k]){
+ datensatz$END_MONATx[k] <- as.numeric(imput_werte10$endmonat18[j])
+ print("Endmonat das imputiert wird")
+ print(datensatz$END_MONATx[k])
+ # Indikator IMP_EM auf 1 setzen, d.h. das Endmonat wurde imputiert
+ datensatz$IMP_EM[k] <- 1
+ }
+ else{
+ if (imput_werte10$endmonat19[j] >= datensatz$ANF_MONAT[k]){
+ datensatz$END_MONATx[k] <- as.numeric(imput_werte10$endmonat19[j])
+ print("Endmonat das imputiert wird")
+ print(datensatz$END_MONATx[k])
+ # Indikator IMP_EM auf 1 setzen, d.h. das Endmonat wurde imputiert
+ datensatz$IMP_EM[k] <- 1
+ }
+ else{
```

```
+ if (imput_werte10$endmonat20[j] >= datensatz$ANF_MONAT[k]){
+ datensatz$END_MONATx[k] <- as.numeric(imput_werte10$endmonat20[j])
+ print("Endmonat das imputiert wird")
+ print(datensatz$END_MONATx[k])
+ # Indikator IMP_EM auf 1 setzen, d.h. das Endmonat wurde imputiert
+ datensatz$IMP_EM[k] <- 1
+ }
+ }}}}}}}}}
+ }
+ }
+ }
+ }
+ }
+ }
+ }
+ }
+ }
+ }
+ }
+ }
+ }
+ j <- j+1
+ }
+
+ # Schauen wo der END_MONATx noch fehlt:
+ endmonatfehlt <- subset(datensatz, is.na(END_MONATx) & kuenstl_geloescht==1)
+ # Für diesen Fall nochmal aus der Verteilung nur bedingt auf SES_r ziehen
+ # da dann ein Anfangsmonat gezogen wurde für den es keine Endmonat >= Anf.monat
+ # gibt
+
+ ############################### SOLAR II #######################################
+ # in SOLAR II bedingen auf SES_r
+ print("**************Imputation des Endmonats in SOLAR II (2)**************")
+ a11 <- subset(datensatz, STUDIE==2 & is.na(END_MONATx) & kuenstl_geloescht == 1)
+ # hier nur die Zeilen drin in denen das END_MONATx künstlich gelöscht wurden
+
+ # Datensatz anlegen, in den jeweils index der Zeile und imputiertes END_MONATx
+ # geschrieben werden (1.Spalte: index, 2.Spalte: wochenstunden)
+ if(nrow(a11)>0){
+ imput_werte11 <- data.frame(index=numeric(nrow(a11)),
+ endmonat1 = numeric(nrow(a11)), endmonat2 = numeric(nrow(a11)),
+ endmonat3 = numeric(nrow(a11)), endmonat4 = numeric(nrow(a11)),
+ endmonat5 = numeric(nrow(a11)), endmonat6 = numeric(nrow(a11)),
+ endmonat7 = numeric(nrow(a11)), endmonat8 = numeric(nrow(a11)),
+ endmonat9 = numeric(nrow(a11)), endmonat10 = numeric(nrow(a11)),
+ endmonat11 = numeric(nrow(a11)), endmonat12 = numeric(nrow(a11)),
+ endmonat13 = numeric(nrow(a11)), endmonat14 = numeric(nrow(a11)),
+ endmonat15 = numeric(nrow(a11)), endmonat16 = numeric(nrow(a11)),
+ endmonat17 = numeric(nrow(a11)), endmonat18 = numeric(nrow(a11)),
+ endmonat19 = numeric(nrow(a11)), endmonat20 = numeric(nrow(a11))
+ )
+
+ # j auf 1 setzen
+ j <- 1
+
+ # Startwert setzen (der der Funktion als Argument übergeben wurde)
+ set.seed(startwert)
+
+ for (i in 1: nrow(a11)){
+ print("Nächste Stelle i")
+ # SES und Index an der i-ten Stelle aus subset betrachten
+ ses <- a11$SES_r[i]
+ print("SES an der Stelle i")
+ print(ses)
+ index <- a11$index[i]
+ print("Index an der Stelle i")
+ print(index)
+ # aus großen Datensatz alle mit gleichem SES ziehen, durch
+ # IMP_EM == 0 werden diejenigen Fälle NICHT zur Berechnung der W.keiten
+ # verwendet, bei denen das END_MONATx bereits imputiert wurde
+ b <- subset(datensatz, SES_r==ses & !is.na(END_MONATx) & STUDIE == 2
+  & IMP_EM==0 )
+ print("Anzahl der Fälle mit gleichem SES")
```

```
+ print(nrow(b))
+
+ table1 <- table(b$END_MONATx)
+ print("Table der Endmonate bedingt auf SES_r")
+ print(table1)
+ prob1 <- prop.table(as.array(table1))  # Wahrscheinlichkeiten für END_MONATx
+ # berechnen
+ print("Wahrscheinlichkeiten der Endmonate bedingt auf Geschlecht")
+ print(prob1)
+
+ # dann aus den Endmonaten mit diesen Wahrscheinlichkeiten ziehen
+ # Endmonat für die Imputation ziehen mit der Funktion sample
+ imput <- as.list(sample(names(prob1), size = 20, replace=TRUE, prob = prob1))
+ # Im Datensatz imputieren, der die Daten aus SOLAR I und SOLAR II enthält
+ print("Endmonat das an dieser Stelle imputiert werden soll")
+ print(imput)
+ # Index der Zeile (index) in 1.Spalte des Datensatzes imput_werte11 speichern
+ imput_werte11$index[j] <- index
+ # Endmonat (imput) in 2.Spalte des Datensatzes imput_werte11 speichern
+ imput_werte11$endmonat1[j] <- as.numeric(imput[[1]])
+ imput_werte11$endmonat2[j] <- as.numeric(imput[[2]])
+ imput_werte11$endmonat3[j] <- as.numeric(imput[[3]])
+ imput_werte11$endmonat4[j] <- as.numeric(imput[[4]])
+ imput_werte11$endmonat5[j] <- as.numeric(imput[[5]])
+ imput_werte11$endmonat6[j] <- as.numeric(imput[[6]])
+ imput_werte11$endmonat7[j] <- as.numeric(imput[[7]])
+ imput_werte11$endmonat8[j] <- as.numeric(imput[[8]])
+ imput_werte11$endmonat9[j] <- as.numeric(imput[[9]])
+ imput_werte11$endmonat10[j] <- as.numeric(imput[[10]])
+ imput_werte11$endmonat11[j] <- as.numeric(imput[[11]])
+ imput_werte11$endmonat12[j] <- as.numeric(imput[[12]])
+ imput_werte11$endmonat13[j] <- as.numeric(imput[[13]])
+ imput_werte11$endmonat14[j] <- as.numeric(imput[[14]])
+ imput_werte11$endmonat15[j] <- as.numeric(imput[[15]])
+ imput_werte11$endmonat16[j] <- as.numeric(imput[[16]])
+ imput_werte11$endmonat17[j] <- as.numeric(imput[[17]])
+ imput_werte11$endmonat18[j] <- as.numeric(imput[[18]])
+ imput_werte11$endmonat19[j] <- as.numeric(imput[[19]])
+ imput_werte11$endmonat20[j] <- as.numeric(imput[[20]])
+ print("Matrix,1.Spalte:Index der Zeile,2.Spalte:Zu imputierendes Endmonat")
+ print(imput_werte11)
+ # Jetzt den Wert im "großen" Datensatz imputieren
+ for(k in 1:nrow(datensatz)){
+ # Wenn der Index übereinstimmt
+ if(datensatz$index[k] == imput_werte11$index[j]){
+ # Zu imputierendes Endmonat in der Zeile mit diesem Index imputieren
+ if (datensatz$ANF_JAHR[k] != datensatz$END_JAHRx[k]){
+ datensatz$END_MONATx[k] <- as.numeric(imput_werte11$endmonat1[j])
+ print("Endmonat das imputiert wird")
+ print(datensatz$END_MONATx[k])
+ # Indikator IMP_EM auf 1 setzen, d.h. das Endmonat wurde imputiert
+ datensatz$IMP_EM[k] <- 1
+ }
+ if (datensatz$ANF_JAHR[k] == datensatz$END_JAHRx[k]){
+ if (imput_werte11$endmonat1[j] >= datensatz$ANF_MONAT[k]){
+ datensatz$END_MONATx[k] <- as.numeric(imput_werte11$endmonat1[j])
+ print("Endmonat das imputiert wird")
+ print(datensatz$END_MONATx[k])
+ # Indikator IMP_EM auf 1 setzen, d.h. das Endmonat wurde imputiert
+ datensatz$IMP_EM[k] <- 1
+ }
+ else{
+ if (imput_werte11$endmonat2[j] >= datensatz$ANF_MONAT[k]){
+ datensatz$END_MONATx[k] <- as.numeric(imput_werte11$endmonat2[j])
+ print("Endmonat das imputiert wird")
+ print(datensatz$END_MONATx[k])
+ # Indikator IMP_EM auf 1 setzen, d.h. das Endmonat wurde imputiert
+ datensatz$IMP_EM[k] <- 1
+ }
+ else{
+ if (imput_werte11$endmonat3[j] >= datensatz$ANF_MONAT[k]){
+ datensatz$END_MONATx[k] <- as.numeric(imput_werte11$endmonat3[j])
+ print("Endmonat das imputiert wird")
```

```
+ print(datensatz$END_MONATx[k])
+ # Indikator IMP_EM auf 1 setzen, d.h. das Endmonat wurde imputiert
+ datensatz$IMP_EM[k] <- 1
+ }
+ else{
+ if (imput_werte11$endmonat4[j] >= datensatz$ANF_MONAT[k]){
+ datensatz$END_MONATx[k] <- as.numeric(imput_werte11$endmonat4[j])
+ print("Endmonat das imputiert wird")
+ print(datensatz$END_MONATx[k])
+ # Indikator IMP_EM auf 1 setzen, d.h. das Endmonat wurde imputiert
+ datensatz$IMP_EM[k] <- 1
+ }
+ else{
+ if (imput_werte11$endmonat5[j] >= datensatz$ANF_MONAT[k]){
+ datensatz$END_MONATx[k] <- as.numeric(imput_werte11$endmonat5[j])
+ print("Endmonat das imputiert wird")
+ print(datensatz$END_MONATx[k])
+ # Indikator IMP_EM auf 1 setzen, d.h. das Endmonat wurde imputiert
+ datensatz$IMP_EM[k] <- 1
+ }
+ else{
+ if (imput_werte11$endmonat6[j] >= datensatz$ANF_MONAT[k]){
+ datensatz$END_MONATx[k] <- as.numeric(imput_werte11$endmonat6[j])
+ print("Endmonat das imputiert wird")
+ print(datensatz$END_MONATx[k])
+ # Indikator IMP_EM auf 1 setzen, d.h. das Endmonat wurde imputiert
+ datensatz$IMP_EM[k] <- 1
+ }
+ else{
+ if (imput_werte11$endmonat7[j] >= datensatz$ANF_MONAT[k]){
+ datensatz$END_MONATx[k] <- as.numeric(imput_werte11$endmonat7[j])
+ print("Endmonat das imputiert wird")
+ print(datensatz$END_MONATx[k])
+ # Indikator IMP_EM auf 1 setzen, d.h. das Endmonat wurde imputiert
+ datensatz$IMP_EM[k] <- 1
+ }
+ else{
+ if (imput_werte11$endmonat8[j] >= datensatz$ANF_MONAT[k]){
+ datensatz$END_MONATx[k] <- as.numeric(imput_werte11$endmonat8[j])
+ print("Endmonat das imputiert wird")
+ print(datensatz$END_MONATx[k])
+ # Indikator IMP_EM auf 1 setzen, d.h. das Endmonat wurde imputiert
+ datensatz$IMP_EM[k] <- 1
+ }
+ else{
+ if (imput_werte11$endmonat9[j] >= datensatz$ANF_MONAT[k]){
+ datensatz$END_MONATx[k] <- as.numeric(imput_werte11$endmonat9[j])
+ print("Endmonat das imputiert wird")
+ print(datensatz$END_MONATx[k])
+ # Indikator IMP_EM auf 1 setzen, d.h. das Endmonat wurde imputiert
+ datensatz$IMP_EM[k] <- 1
+ }
+ else{
+ if (imput_werte11$endmonat10[j] >= datensatz$ANF_MONAT[k]){
+ datensatz$END_MONATx[k] <- as.numeric(imput_werte11$endmonat10[j])
+ print("Endmonat das imputiert wird")
+ print(datensatz$END_MONATx[k])
+ # Indikator IMP_EM auf 1 setzen, d.h. das Endmonat wurde imputiert
+ datensatz$IMP_EM[k] <- 1
+ }
+ else{
+ if (imput_werte11$endmonat11[j] >= datensatz$ANF_MONAT[k]){
+ datensatz$END_MONATx[k] <- as.numeric(imput_werte11$endmonat11[j])
+ print("Endmonat das imputiert wird")
+ print(datensatz$END_MONATx[k])
+ # Indikator IMP_EM auf 1 setzen, d.h. das Endmonat wurde imputiert
+ datensatz$IMP_EM[k] <- 1
+ }
+ else{
+ if (imput_werte11$endmonat12[j] >= datensatz$ANF_MONAT[k]){
+ datensatz$END_MONATx[k] <- as.numeric(imput_werte11$endmonat12[j])
+ print("Endmonat das imputiert wird")
+ print(datensatz$END_MONATx[k])
```

```
+ # Indikator IMP_EM auf 1 setzen, d.h. das Endmonat wurde imputiert
+ datensatz$IMP_EM[k] <- 1
+ }
+ else{
+ if (imput_werte11$endmonat13[j] >= datensatz$ANF_MONAT[k]){
+ datensatz$END_MONATx[k] <- as.numeric(imput_werte11$endmonat13[j])
+ print("Endmonat das imputiert wird")
+ print(datensatz$END_MONATx[k])
+ # Indikator IMP_EM auf 1 setzen, d.h. das Endmonat wurde imputiert
+ datensatz$IMP_EM[k] <- 1
+ }
+ else{
+ if (imput_werte11$endmonat14[j] >= datensatz$ANF_MONAT[k]){
+ datensatz$END_MONATx[k] <- as.numeric(imput_werte11$endmonat14[j])
+ print("Endmonat das imputiert wird")
+ print(datensatz$END_MONATx[k])
+ # Indikator IMP_EM auf 1 setzen, d.h. das Endmonat wurde imputiert
+ datensatz$IMP_EM[k] <- 1
+ }
+ else{
+ if (imput_werte11$endmonat15[j] >= datensatz$ANF_MONAT[k]){
+ datensatz$END_MONATx[k] <- as.numeric(imput_werte11$endmonat15[j])
+ print("Endmonat das imputiert wird")
+ print(datensatz$END_MONATx[k])
+ # Indikator IMP_EM auf 1 setzen, d.h. das Endmonat wurde imputiert
+ datensatz$IMP_EM[k] <- 1
+ }
+ else{
+ if (imput_werte11$endmonat16[j] >= datensatz$ANF_MONAT[k]){
+ datensatz$END_MONATx[k] <- as.numeric(imput_werte11$endmonat16[j])
+ print("Endmonat das imputiert wird")
+ print(datensatz$END_MONATx[k])
+ # Indikator IMP_EM auf 1 setzen, d.h. das Endmonat wurde imputiert
+ datensatz$IMP_EM[k] <- 1
+ }
+ else{
+ if (imput_werte11$endmonat17[j] >= datensatz$ANF_MONAT[k]){
+ datensatz$END_MONATx[k] <- as.numeric(imput_werte11$endmonat17[j])
+ print("Endmonat das imputiert wird")
+ print(datensatz$END_MONATx[k])
+ # Indikator IMP_EM auf 1 setzen, d.h. das Endmonat wurde imputiert
+ datensatz$IMP_EM[k] <- 1
+ }
+ else{
+ if (imput_werte11$endmonat18[j] >= datensatz$ANF_MONAT[k]){
+ datensatz$END_MONATx[k] <- as.numeric(imput_werte11$endmonat18[j])
+ print("Endmonat das imputiert wird")
+ print(datensatz$END_MONATx[k])
+ # Indikator IMP_EM auf 1 setzen, d.h. das Endmonat wurde imputiert
+ datensatz$IMP_EM[k] <- 1
+ }
+ else{
+ if (imput_werte11$endmonat19[j] >= datensatz$ANF_MONAT[k]){
+ datensatz$END_MONATx[k] <- as.numeric(imput_werte11$endmonat19[j])
+ print("Endmonat das imputiert wird")
+ print(datensatz$END_MONATx[k])
+ # Indikator IMP_EM auf 1 setzen, d.h. das Endmonat wurde imputiert
+ datensatz$IMP_EM[k] <- 1
+ }
+ else{
+ if (imput_werte11$endmonat20[j] >= datensatz$ANF_MONAT[k]){
+ datensatz$END_MONATx[k] <- as.numeric(imput_werte11$endmonat20[j])
+ print("Endmonat das imputiert wird")
+ print(datensatz$END_MONATx[k])
+ # Indikator IMP_EM auf 1 setzen, d.h. das Endmonat wurde imputiert
+ datensatz$IMP_EM[k] <- 1
+ }
+ }}}}}}}}}
+ }
+ }
+ }
+ }
+ }
```

```
+ }
+ }
+ }
+ }
+ }
+ }
+ }
+ }
+ j <- j+1
+ }
+ }
+
+ ##############################################################################
+ ##################  EXPOSITION imputieren ####################################
+ ##############################################################################
+
+ #datensatz$IMP_EXPO <- 0
+
+ ################################ SOLAR-I ######################################
+ #print("***************Imputation der Exposition in SOLAR I***************")
+
+ # Daten rausziehen, bei denen imputiert werden muss (kuenstl_geloescht == 1),nur
+ # SOLAR I (STUDIE=1) und ISCO fehlt (da Expo nur fehlt wenn ISCO fehlt)
+ #EXPO_s1 <-subset(datensatz_s1, is.na(ISCO) & kuenstl_geloescht == 1)
+
+ #if(nrow(EXPO_s1)>0){
+ #j <- 1
+ #set.seed(startwert)
+ #for(i in 1:nrow(EXPO_s1)){
+ #index <- EXPO_s1$index[i]
+ #print("Index an der Stelle i")
+ #print(index)
+ #expo1 <- subset(datensatz_s1, select=c(10:31), ISCO != 8888 & ISCO != 9999 &
+ #ISCO != 94 & ISCO != 95 & ISCO != 97 & ISCO != 98)
+ #expo1$muster <- ""
+
+ #for (i in 1:nrow(expo1)){
+ #expo1$muster[i] <- paste(expo1[i,],sep="",collapse="")
+ #}
+
+ #table_EXPO_s1 <- table(expo1$muster)
+ #table_EXPO_s1
+ #print("Table der Expositionsmuster")
+ #print(table_EXPO_s1)
+ #prop_EXPO_s1 <- prop.table(as.array(table_EXPO_s1))
+ #prop_EXPO_s1
+ #print("Table der Wahrscheinlichkeiten der Expositionsmuster")
+ #print(prop_EXPO_s1)
+
+ #EXPO_s1_imput <- sample(names(prop_EXPO_s1), size = nrow(EXPO_s1),
+ #replace=TRUE, prob = prop_EXPO_s1)
+ #table(EXPO_s1_imput)
+ #print("Expositionsmuster das imputiert wird")
+ #print(EXPO_s1_imput)
+
+ # Expo im Datensatz ersetzen
+ #j <- 1
+ #for (i in 1:nrow(datensatz)){
+ #  if(is.na(datensatz$ISCO)[i] &
+ #  datensatz$kuenstl_geloescht[i] == 1 & datensatz$STUDIE[i] == 1){
+ #    datensatz$anim[i] <- as.numeric(substring(EXPO_s1_imput,1,1))
+ #    datensatz$fish[i] <- as.numeric(substring(EXPO_s1_imput,2,2))
+ #    datensatz$flour[i] <- as.numeric(substring(EXPO_s1_imput,3,3))
+ #    datensatz$plants[i] <- as.numeric(substring(EXPO_s1_imput,4,4))
+ #    datensatz$mites[i] <- as.numeric(substring(EXPO_s1_imput,5,5))
+ #    datensatz$enzymes[i] <- as.numeric(substring(EXPO_s1_imput,6,6))
+ #    datensatz$latex[i] <- as.numeric(substring(EXPO_s1_imput,7,7))
+ #    datensatz$bioaero[i] <- as.numeric(substring(EXPO_s1_imput,8,8))
+ #    datensatz$drugs[i] <- as.numeric(substring(EXPO_s1_imput,9,9))
+ #    datensatz$react[i] <- as.numeric(substring(EXPO_s1_imput,10,10))
+ #    datensatz$isocy[i] <- as.numeric(substring(EXPO_s1_imput,11,11))
+ #    datensatz$clean[i] <- as.numeric(substring(EXPO_s1_imput,12,12))
+ #    datensatz$wood[i] <- as.numeric(substring(EXPO_s1_imput,13,13))
```

```
+ #    datensatz$metals[i] <- as.numeric(substring(EXPO_s1_imput,14,14))
+ #    datensatz$mwf[i] <- as.numeric(substring(EXPO_s1_imput,15,15))
+ #    datensatz$textile[i] <- as.numeric(substring(EXPO_s1_imput,16,16))
+ #    datensatz$agric[i] <- as.numeric(substring(EXPO_s1_imput,17,17))
+ #    datensatz$irrpeaks[i] <- as.numeric(substring(EXPO_s1_imput,18,18))
+ #    datensatz$exhaust[i] <- as.numeric(substring(EXPO_s1_imput,19,19))
+ #    datensatz$ets[i] <- as.numeric(substring(EXPO_s1_imput,20,20))
+ #    datensatz$pos_irr[i] <- as.numeric(substring(EXPO_s1_imput,21,21))
+ #    datensatz$low_anti[i] <- as.numeric(substring(EXPO_s1_imput,22,22))
+ #    datensatz$IMP_EXPO[i] <- 1
+ #    j <- j+1
+ #  }
+ #}
+ #}
+ #}
+
+
+ ################################ SOLAR-II ###################################
+ #print("***************Imputation der Exposition in SOLAR II***************")
+ # Daten rausziehen, bei denen imputiert werden muss (kuenstl_geloescht == 1),nur
+ # SOLAR II (STUDIE=2) und ISCO fehlt (da Expo nur fehlt wenn ISCO fehlt)
+ #EXPO_s2 <-subset(datensatz_s2, is.na(ISCO) & kuenstl_geloescht == 1)
+ #if(nrow(EXPO_s2)>0){
+ #j <- 1
+ #set.seed(startwert)
+ #for(i in 1:nrow(EXPO_s2)){
+ #index <- EXPO_s2$index[i]
+ #print("Index an der Stelle i")
+ #print(index)
+ #expo2 <- subset(datensatz_s2, select=c(10:31), ISCO != 8888 & ISCO != 9999 &
+ #ISCO != 94 & ISCO != 95 & ISCO != 97 & ISCO != 98)
+ #expo2$muster <- ""
+
+ #for (i in 1:nrow(expo2)){
+ #expo2$muster[i] <- paste(expo2[i,],sep="",collapse="")
+ #}
+
+ #table_EXPO_s2 <- table(expo2$muster)
+ #table_EXPO_s2
+ #print("Table der Expositionsmuster")
+ #print(table_EXPO_s2)
+ #prop_EXPO_s2 <- prop.table(as.array(table_EXPO_s2))
+ #prop_EXPO_s2
+ #print("Table der Wahrscheinlichkeiten der Expositionsmuster")
+ #print(prop_EXPO_s2)
+
+
+ #EXPO_s2_imput <- sample(names(prop_EXPO_s2), size = nrow(EXPO_s2),
+ #replace=TRUE, prob = prop_EXPO_s2)
+ #table(EXPO_s2_imput)
+ #print("Expositionsmuster das imputiert wird")
+ #print(EXPO_s2_imput)
+
+ # Expo im Datensatz ersetzen
+ #j <- 1
+ #for (i in 1:nrow(datensatz)){
+ #  if(is.na(datensatz$ISCO)[i] &
+ #  datensatz$kuenstl_geloescht[i] == 1 & datensatz$STUDIE[i] == 2){
+ #    datensatz$anim[i] <- as.numeric(substring(EXPO_s2_imput,1,1))
+ #    datensatz$fish[i] <- as.numeric(substring(EXPO_s2_imput,2,2))
+ #    datensatz$flour[i] <- as.numeric(substring(EXPO_s2_imput,3,3))
+ #    datensatz$plants[i] <- as.numeric(substring(EXPO_s2_imput,4,4))
+ #    datensatz$mites[i] <- as.numeric(substring(EXPO_s2_imput,5,5))
+ #    datensatz$enzymes[i] <- as.numeric(substring(EXPO_s2_imput,6,6))
+ #    datensatz$latex[i] <- as.numeric(substring(EXPO_s2_imput,7,7))
+ #    datensatz$bioaero[i] <- as.numeric(substring(EXPO_s2_imput,8,8))
+ #    datensatz$drugs[i] <- as.numeric(substring(EXPO_s2_imput,9,9))
+ #    datensatz$react[i] <- as.numeric(substring(EXPO_s2_imput,10,10))
+ #    datensatz$isocy[i] <- as.numeric(substring(EXPO_s2_imput,11,11))
+ #    datensatz$clean[i] <- as.numeric(substring(EXPO_s2_imput,12,12))
+ #    datensatz$wood[i] <- as.numeric(substring(EXPO_s2_imput,13,13))
+ #    datensatz$metals[i] <- as.numeric(substring(EXPO_s2_imput,14,14))
+ #    datensatz$mwf[i] <- as.numeric(substring(EXPO_s2_imput,15,15))
```

```
+ #    datensatz$textile[i] <- as.numeric(substring(EXPO_s2_imput,16,16))
+ #    datensatz$agric[i] <- as.numeric(substring(EXPO_s2_imput,17,17))
+ #    datensatz$irrpeaks[i] <- as.numeric(substring(EXPO_s2_imput,18,18))
+ #    datensatz$exhaust[i] <- as.numeric(substring(EXPO_s2_imput,19,19))
+ #    datensatz$ets[i] <- as.numeric(substring(EXPO_s2_imput,20,20))
+ #    datensatz$pos_irr[i] <- as.numeric(substring(EXPO_s2_imput,21,21))
+ #    datensatz$low_anti[i] <- as.numeric(substring(EXPO_s2_imput,22,22))
+ #    datensatz$IMP_EXPO[i] <- 1
+ #    j <- j+1
+ #  }
+ #}
+ #}
+ #}
+ # Datensatz der jetzt die imputierten Werte enthält soll zurückgegeben werden
+ return(datensatz)
+ }
```

ANHANG D

CD Inhalt

Die beiliegende CD enthält neben der digitalen Ausgabe der vorliegenden Arbeit die in dieser Arbeit zitierte Literatur, die gesamten R-Dokumente sowie Material zu den behandelten Studien (Fragebögen, Kodierungsmanuals).

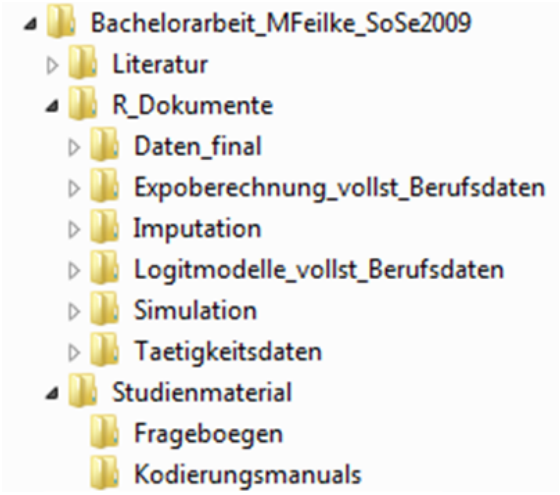


Abbildung D.1: Ordnerstruktur der CD

Literaturverzeichnis

[Anderson et al. 1992] Anderson, H., A. Pottier und D. Strachan (1992). *Asthma from birth to age 23: incidence and relation to prior and concurrent atopic disease*. Thorax, 47, 537-542.

[Benke et al. 2008] Benke, G., M. Sim, D. McKenzie, E. MacFarlane, A. D. Monaco, J. Hoving und L. Fritschi (2008). *Comparison of First, Last and Longest-Held Jobs as Surrogates for All Jobs in Estimating Cumulative Exposure in Cross-Sectional Studies of Work-Related Asthma*. Annals of Epidemiology, 18, 23-27.

[Fahrmeir et al. 2007] Fahrmeir, L., T. Kneib und S. Lang (2007). *Regression - Modelle, Methoden und Anwendungen*. Springer Verlag, 5 Aufl.

[Fahrmeir et al. 2004] Fahrmeir, L., R. Künstler, I. Pigeot und G. Tutz (2004). *Statistik - Der Weg zur Datenanalyse*. Springer Verlag, 5 Aufl.

[Geis 2007] Geis, A. (2007). *Handbuch für die Berufsvercodung*. ZUMA Zentrum für Umfragen, Methoden und Analysen.

[Honaker und King 2008] Honaker, J. und G. King (2008). *What to do about Missing Values in Time Series Cross-Section Data*. http://gking.harvard.edu/files/pr.pdf.

[Honaker et al. 2009] Honaker, J., G. King und M. Blackwell (2009). *Amelia II: A Program for Missing Data - Version 1.2-0*. http://gking.harvard.edu/amelia/docs/amelia.pdf.

[Kennedy et al. 2000] Kennedy, S., N. LeMounal, D. Choudat und F. Kauffmann (2000). *Development of an asthma specific job exposure matrix and its application in the epidemiological study of genetics and environment in asthma (EGEA)*. Occupational and Environmental Medicine, 57, 635-641.

[King et al. 2001] King, G., J. Honaker, A. Joseph und K. Scheve (2001). *Analyzing Incomplete Political Science Data: An Alternative Algorithm for Multiple Imputation*. American Political Science Review, 95.

[Little 2002] Little, R. & Rubin, D. (2002). *Statistical Analysis with Missing Data*. Wiley-Interscience, 2 Aufl.

[Radon 2005] Radon, K. et al. (2005). *Berufliche Allergierisiken - Die SOLAR-Kohortenstudie*. Schriftenreihe der Bundesanstalt für Arbeitsschutz und Arbeitsmedizin.

[Radon 2008] Radon, K. et al. (2008). *Manifestation allergischer Krankheiten bei jungen Erwachsenen in Zusammenhang mit dem Eintritt in das Berufsleben*. Sachstandsbericht November 2008.

[Riu et al. 2007] Riu, E., H. Dressel, D. Windstetter, G. Weinmayr, S. Weiland, C. Vogelberg, W. Leupold, E. von Mutius, D.Nowak und K. Radon (2007). *First months of employment and new onset of rhinitis in adolescents*. Eur Respir J, 30, 549-555.

[Sachs und Hedderich 2006] Sachs, L. und J. Hedderich (2006). *Angewandte Statistik - Methodensammlung mit R*. Springer Verlag, 12 Aufl.

[Schafer und Olsen 1998] Schafer, J. und M. Olsen (1998). *Multiple Imputation for Multivariate Missing-Data Problems: A Data Analyst´s Perspective*. Multivariate Behavioral Research, 33(4), 545-571.

[Strachan 1989] Strachan, D. (1989). *Hay fever, hygiene, and household size*. BMJ, 299, 1259-1260.

[Toutenburg 2003] Toutenburg, H. (2003). *Lineare Modelle*. Physica Verlag, 2 Aufl.

[Toutenburg und Heumann 2006] Toutenburg, H. und C. Heumann (2006). *Deskriptive Statistik - Eine Einführung in Methoden und Anwendungen mit SPSS*. Springer Verlag, 5 Aufl.